AD-A039 394 UNIVERSITY COLL CORK (IRELAND) DEPT OF MATHEMATICAL--ETC F/G 12/1
FUNDAMENTALS OF THE EDGE-FUNCTION METHOD VIA LAPLACE'S EQUATION--ETC(U)
FEB 77 P M QUINLAN AF-AFOSR-2786-75

UNCLASSIFIED EOARD-TR-77-01 NL

| OF |

AD
A039 394

END

DATE
FILMED
6-77

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: EOARD-TR-77-01	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Fundamentals of The Edge-Function Method via Laplace's Equation.	5. TYPE OF REPORT & PERIOD COVERED: Scientific Interim rept. 1 July 1975 - 30 June 1976
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Patrick M. Quinlan	8. CONTRACT OR GRANT NUMBER(s): AF-AFOSR-75-2786-75
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University College of Cork, Dept. of Mathematical Physics, Cork, Ireland.	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 9767-99-56
11. CONTROLLING OFFICE NAME AND ADDRESS: European Office of Aerospace, Research & Development, FPO N.Y. 09510	12. REPORT DATE: 1 February, 1977
	13. NUMBER OF PAGES: 50 + 2 progs
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): European Office of Aerospace Research & Development FPO N.Y. 09510	15. SECURITY CLASS. (of this report)
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Edge-Function Method, Structural Analysis, Linear Boundary Value Problems, Laplace's Equation; Harmonic Fitting

the Edge-Function Method

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The main algebraic and programming features of E.F.M. are illustrated by application to Laplace's equation for polygonal regions with elliptical cavities. The Basic Functions - Vertex, Edge and Mapped Polars - are obtained. Vertex Functions, Harmonic Fitting and submission into elements are covered. Programs are attached.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

402 948

FUNDAMENTALS OF THE EDGE-FUNCTION METHOD VIA LAPLACE'S EQUATION

by

P. M. QUINLAN(1)

Introduction

1. OUTLINE

The Edge-Function Method as applied to two-dimensional problems is a procedure for obtaining approximate solutions to boundary value problems in regular or irregular polygonal regions with, or without, cavities and cracks. It may be described as a piecing together of "asymptotic" solutions to a set of linear partial differential equations for the several parts of a domain D to satisfy the boundary conditions in a discrete least squares sense.

The method was originally developed by Quinlan [1] for the solution of the torsion problem for prismatic bars of polygonal cross-section. In the succeeding 10 years it has been successfully applied to problems in the bending of isotropic thin plates [2,3,4]; coupled linear systems in elastostatics [4,5,6,7] and in moderately thick plates and shallow shells [8]; cracks and stress concentrations in elastostatics [9]; vibrations of Thin Plates [10] and vibrations of Shallow Shells [11].

The present paper arose out of a course of lectures on E.F.M. given by the author in December 1975 at The Georgia Institute of Technology. The aim is to illustrate as simply as possible the main algebraic and programming features of the method by applying it to Laplace's equation as it arises in heat flow and torsion problems.

2. PLAN OF PAPER

The paper is divided into four sections:

(a) Section 1: Introduces Edge-Functions and Vertex Functions for steady state heat flow in a polygonal region the temperatures being specified on the boundary. The resulting boundary identities are satisfied by harmonic matching using Fourier sine series.

The importance of the vertex equations is shown. The Edge-Functions are obtained using a half-strip mathematical model.

A simple program, LAPEX, and an illustrative example, is given in Appendix A for problems covered by this section.

---

(1) Professor of Mathematical Physics, University College, Cork, Ireland.

(b) Section 2: Extension is made to regions with mixed conditions on the boundary segments,and circular holes are included. Harmonic matching is now based on using the (full) Fourier series, with corresponding (full) Edge-Functions.

Provision is made for Logarithmic Singularities, where they arise, using Log-Vertex Functions.

Programming requirements for a comprehensive program LAPGEN, as a development of LAPEX, are discussed and a suitable program LAPGEN, with examples, is attached in Appendix B.

(c) Section 3 - Curved Boundaries: This uses conformal mapping to extend the treatment of the previous section to holes of elliptic shape, and to circular and/or elliptic indentations on the boundary. The ensuing Curved Edge-Functions are developed, and programmed in subroutine PMAP in LAPGEN for elliptic curves.

(d) Section 4 - Other Basic Problems: This section treats the case of a reentrant angle, and also where singularities occur on the boundary, using appropriate half-strip models. These features have not been included in LAPGEN, but are available in QUINP.

The models used may be termed "BASIC MODELS" for the problem, the corresponding regions being termed "BASIC REGIONS" and the corresponding functions being termed "BASIC FUNCTIONS".

(e) Section 5 - Harmonic Fitting: A discrete least squares method for fitting a set of orthogonal functions to a given curve is developed. It is applied to solve the boundary identity problem by minimising the boundary residuals, on each segment, using a discrete least squares criterion. This process is given the name "Harmonic Fitting", and follows from Harmonic Matching when the integrals concerned are replaced by the corresponding trapezoidal rule quadrature formula.

(f) Section 6 - Distinctive Features of the Edge Function Method.

Appendix A: LAPEX, with example of heat flow in a quadrilateral.

Appendix B: LAPGEN with example involving the torsion of a quadilateral region with an elliptic hole

Appendix C: Data Input for LAPGEN on the "black box" principle.

SUMMARY OF METHOD

Accordingly to apply The Edge-Function Method to any new situation e.g. thin plates, shells, 3-D Elasticity, we require to find the corresponding BASIC FUNCTIONS, and fit a superposition of these to the boundary conditions by "HARMONIC FITTING" using a computation scheme similar to that given in the attached program LAPGEN.

Sections 3, 4 and 5 may be omitted when studying the Edge-Function Method for the first time.

SECTION 1

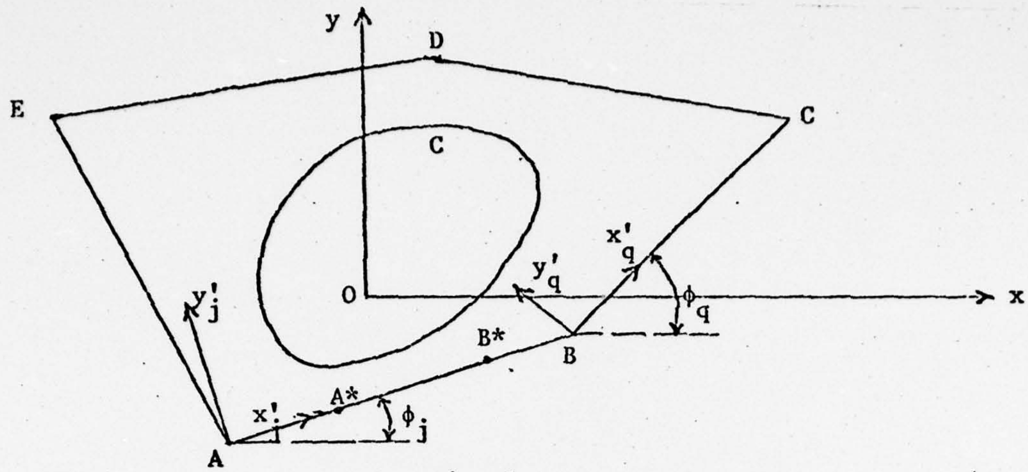


Fig. 1

Consider the linear partial differential equation:

$$L(u) = f(x,y); \quad L \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \tag{1}$$

where L, in the present paper, is restricted to the Laplace operator.

It is required to find a function $u = u(x,y)$ which satisfies the above equation in a domain Ω bounded by any number q' piecewise smooth segments, or straight lines, and which takes prescribed values

$$u = g_q(x,y) \tag{2}$$

on the various boundary segments corresponding to $q = 1,2,\ldots,q'$.

For example in a steady state heat flow problem u is the temperature and $f(x,y)$ is the heat generation term, and for a heat source of strength Q at a point (u,v), f is given by

$$f(x,y) = Q\delta(x-u,\ y-v), \tag{3}$$

where δ is the delta function.

If $u = u^p$ is a particular solution of eqn.(1), then

$$L(u^p) = f(x,y) \tag{4}$$

A more general solution, involving the addition of another function u^c, may be written in the form

$$u = u^p + u^c \tag{5}$$

where, since L is linear,

$$L(u^p + u^c) = L(u^p) + L(u^c) = f(x,y), \tag{6}$$

and on subtracting eqn.(4), it follows that

$$L(u^c) = 0, \tag{7}$$

which is called the complementary equation of eqn.(1).

The problem is now split into finding

(a) a particular solution u^p, and

(b) suitable complementary functions u^c, with sufficient flexibility to enable solution $u^p + u^c$ to satisfy, in an approximate manner, the boundary conditions (2).

The present paper is restricted to the case $f = 0$, for which $u^p = 0$.

On obtaining I suitable solutions u_i^c, $i = 1,I$; of eqn.(7) and superposing, we obtain

$$u = \sum_{i=1}^{I} A_i u_i^c \tag{8}$$

involving the I arbitrary superposition constants A_i, and these must be evaluated to satisfy conditions (2) in an acceptable approximate manner.

I eqns. are required to determine the I constants A_i. Various schemes for producing I eqns. are:

(a) Point-Matching, or collocation [Conway, Leissa etc.]
Select I points on boundary in same manner - equidistant distribution, or have a somewhat closer spacing near corners— and write eqns. to ensure that boundary conditions (2) are satisfied at the selected points. Check the boundary residuals (or deviations) at points in between the selected points. The system of simultaneous equations may become unstable, giving ill-conditioned equations, as I is increased. There is an art in selecting the optimum spacing for the points.

(b) Least squares fitting This requires that the boundary conditions be satisfied, in a least squares sense, at M points, where M > I, there being more points selected than there are unknowns. This improves the process up to about $M \approx 2I$, but much additional computation is involved in producing the resulting normalised equations.

(c) Harmonic Matching (Quinlan 1962-70)
Consider the boundary identity on the q^{th} side, in fig.1, where points on the side are specified by the parameter x_q', which requires, on substituting from eqn.(8) in (2), that

$$\sum_{i=1}^{I} A_i u_i^c \equiv g_q(x,y), \tag{9}$$

for all points $P(x_q')$ on q^{th} side in the range $(0,a_q)$.
Since the coordinates (x,y) for any point P can be expressed in terms of x_q', identity (9) can be expressed as

$$\sum_{i=1}^{I} A_i u_i^c(x_q') - g_q(x_q') \equiv 0, \tag{10}$$

or, for short,

$$\psi_q(x_q') \equiv 0; \quad 0 \leq x_q' \leq a_q, \tag{11}$$

with similar identities for each boundary segment.

On expanding identity (11) in a Fourier sine series, we obtain

$$\sum_{N=1}^{\infty} C_N \sin nx_q' \equiv 0; \quad n = \frac{N\pi}{a_q}, \tag{12}$$

where the Fourier coefficients C_N are given by the integral

$$C_N = \frac{2}{a_q} \int_o^{a_q} \psi_q(x_q') \sin nx_q' \, dx_q' \tag{13}$$

Since we have but I degrees of freedom in eqn.(8) - or I arbitrary constants A_i - and if these were to be shared out equally between the q' boundary segments, we would then allocate N^*, where

$$N^* = I/q' \tag{14}$$

to each boundary identity (11). Accordingly, the best way to satisfy identity (12) would appear to be to set the first N^* harmonics to zero, giving rise to the equations

$$\begin{aligned} C_1 &= 0 \\ C_2 &= 0 \\ &\cdot \\ &\cdot \\ &\cdot \\ C_{N^*} &= 0 \end{aligned} \tag{15}$$

<u>This process is called Harmonic Matching</u>

On substituting from eqs.(11) and (10) into (13), we obtain, on setting $C_N = 0$, for the Nth harmonic:

$$\sum_{i=1}^{I} A_i \int_o^{a_q} u_i^c(x_q') \sin nx_q' \, dx_q' - \int_o^{a_q} g_q(x_q') \sin nx_q' \, dx_q' = 0, \tag{16}$$

where N goes from 1 to N^*. On evaluating, a linear equation results in the unknowns A_i. Similar linear equations follow for each side of the polygon and accordingly a total of I linear equations result for the I unknowns, A_i, which system may be written in the matrix form

$$\tilde{P}\bar{x} = \bar{Q}, \tag{17}$$

where $\tilde{P}$ is the I x I coefficients matrix.

However setting the first N^* harmonics to zero does not satisfy (12) exactly, but leaves as a residual the higher harmonics

$$\sum_{N=\bar{N}}^{\infty} C_N \sin nx'_q; \quad \bar{N} = N^* + 1 \tag{18}$$

and we require this to be negligible - which will be the case only if the series is sufficiently convergent.

As shown in texts on Fourier series the sine series expansion of (11) will have a Gibbs effect at the ends $x'_q = 0$; $x'_q = a_q$ - with convergence only of order $\frac{1}{M}$ - unless the function being expanded, ψ_q, is zero at both ends of the range. In this latter case the convergence improves considerably, to be of order $1/M^3$, and consequently the residual series (18) can be made negligible, in a practical sense, by taking N^* sufficiently large. Accordingly we must ensure in identity (11) that

$$\psi_q(x'_q) = 0; \quad x'_q = 0 \text{ and } x'_q = a_q \tag{19}$$

This involves two additional equations for each side - called the Vertex Equations - and if we still wish to restrict to N^* equations per side, we must reduce the harmonic equations set (15) for each side by two, corresponding to truncation at L harmonics where $L = N^*-2$. It is obvious, due to the greatly increased convergence of the expansion (12) to $1/M^3$, that this scheme is a better utilisation of the available resources per side (N^* constants) in seeking to satisfy, in an acceptable approximate way, the boundary identities (11).

If the temperature is continuous at all points on the boundary, including all corner points, then eqs.(19) reduce to a single equation $\psi = 0$ at each vertex.

Basic Problems

If solution form (8) is capable of providing an acceptable solution to the problem, then it is especially important that it should represent the main features of the solution in all the critical regions of Fig. 1. On examining some critical parts of the boundary we note that:

(a) in any corner, say at A, the solution (1-8) must approach the solution in an infinite sector B'AE', where B' and E' are points at infinity on AB and AE respectively. In particular, if there is any singular behaviour - physically corresponding to infinite stresses - in the infinite sector solution, there must be corresponding singularities in solution (1-8). Hence we must first solve the infinite sector problem, and include its more singular terms - called Vertex Functions - in superposition (8) for each of the vertices in Figure 1.

(b) at a distance from any corner where the vertex effects have moderated, say on A*B*, solution (8) should, in the neighbourhood of A*B*, behave like the solution for the half-plane problem bounded by A'B', where A' and B' are points at infinity on AB. Such solutions are termed Edge-Functions, and an appropriate number of these must be included for each straight side of Figure 1.

(c) in the vicinity of interior hole C the solution should approach that of a hole C in an infinite region, and again an appropriate number of the more singular (or characteristic) of these solutions should be included.

The appropriate "mix" of functions from (a), (b) and (c) will be determined later from analytic considerations. We now proceed to solve the basic problems for Laplace's equation, as exemplified by the heat flow problem:-

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \qquad (21)$$

where the superscript c has been omitted.

[a] <u>VERTEX FUNCTIONS</u> (Angle Problem)

Polar type solutions of Laplace's equation are provided by the well known harmonics of order λ:

$$u = Ar^{\lambda} \cos\lambda\theta + Br^{\lambda} \sin\lambda\theta, \tag{21a}$$

where A and B are arbitrary constants, and λ is arbitrary. The appropriate boundary conditions for a sector of angle α, with one side $\theta = 0$, are the corresponding zero conditions,

$$\begin{aligned} u = 0 \; ; \quad \theta = 0 \\ u = 0 \; ; \quad \theta = \alpha, \end{aligned} \tag{22}$$

as these give rise to an eigenvalue problem.

On applying conditions (22) to eqn.(21a) we obtain

$$\begin{aligned} 0 &= A \\ 0 &= Br^{\lambda} \sin\lambda\alpha, \end{aligned}$$

a solution to which, other than the trivial one $u \equiv 0$, is given by

$$\sin\lambda\alpha = 0 \tag{23}$$

This is called the eigenvalue equation for the problem, and its solution is

$$\lambda\alpha = k\pi,$$

where k is any integer. On denoting the above discrete set of values - called eigenvalues - for λ by λ_k we obtain as solutions

$$u_k = Br^{\lambda_k} \sin\lambda_k\theta; \quad \lambda_k = k\pi/\alpha, \tag{24}$$

where u_k are called the eigenfunctions for the problem.

Similar eigenfunctions* - here called <u>Vertex Functions</u> - must be incorporated in "mix" (8) from all vertices $j = 1,j'$; $(j'=q')$. Accordingly the k^{th} eigenfunction, or vertex function, for the j^{th} vertex is denoted by the symbol V_{kj}, where

$$V_{kj} \equiv u_k = B_{kj}[r^{\lambda_k} \sin\lambda_k\theta]_j \tag{25}$$

the subscript j outside the square bracket denotes that the corresponding origin of coordinates is at vertex j, with $\theta = 0$ along side j. The corresponding arbitrary constant is written as B_{kj}, since it can change with k and j.

Note that V_{kj} is a point-function, and this can be emphasised, when required, by writing it as $V_{kj}(P)$.

---

*Eqn.(20) is invariant under a rotation of axes, and accordingly on taking the polar axis $\theta_j = 0$ along the j^{th} side with pole at vertex j, harmonic solutions (21), and eigenfunctions (24) apply to the vertex j.

The permissable range of λ_k values is determined by the physics of the problem. Negative values of λ_k cannot be admitted as these would require infinite temperatures at the corresponding vertices, but can infinite temperature derivatives (at $r_j = 0$) be admitted? Consider the heat flux $\bar{q}$, where

$$\bar{q} = -\beta\nabla u_k = -\beta\{\frac{\partial u}{\partial r}\hat{r} + \frac{1}{r}\frac{\partial u}{\partial \theta}\hat{\theta}\} \tag{26}$$

The flow through any circular arc of angle α, due to each V_{kj}, is given by

$$\begin{aligned} q_j &= \int_0^\alpha \frac{\partial V_{kj}}{\partial r} r d\theta \\ &= \lambda_k r^{\lambda_k} \int_0^\alpha \sin\lambda_k \theta d\theta \\ &= r^{\lambda_k}\{1 - \cos\lambda_j \alpha\}, \end{aligned} \tag{27}$$

and this is infinite - and thus physically inadmissable - if $\lambda_k < 0$. Accordingly in (25) we require that

$$\lambda_k > 0. \tag{28}$$

The corresponding heat flux, as given by (26) on substituting for V_{kj}, is of order $r^{\lambda_{k-1}}$, and if $0 < \lambda_k < 1$ the flux is infinite at $r = 0$, but this is admissable since the nett heat flow through the vertex is zero.

Note that, if λ_k is not an integer, all roots $\lambda_k > 0$ have infinite derivatives of order p and higher, where $\lambda_k - p < 0$. Accordingly all functions V_{kj} are singularity functions in the sense that all their derivatives above a certain order are singular at $r = 0$.

When a temperature discontinuity occurs at any point P on the boundary, or at any vertex, the corresponding discontinuity can be incorporated in "mix" (8) by including an zero order harmonic solution to eqn.(20) for each such point given by

$$u_o = B_o\theta \tag{29}$$

[b] EDGE FUNCTIONS (half-plane problem)

Consider the solution in half plane $y'_j \geq 0$, where the edge-axes for j^{th} side denoted (x'_j, y'_j) are taken respectively along and perpendicular to the j^{th} side and are at an angle ϕ_j to the reference axis (x,y).

On transforming eqn.(7) to the above edge-axes with the aid of the relations:

$$\begin{aligned} x'_j &= (x-x_j)\cos\phi_j + (y-y_j)\sin\phi_j \\ y'_j &= -(x-x_j)\sin\phi_j + (y-y_j)\cos\phi_j \\ \frac{\partial}{\partial x} &= \hat{x}.\nabla = \hat{x}.(\hat{x}'_j \frac{\partial}{\partial x'_j} + \hat{y}'_j \frac{\partial}{\partial y'_j}) \\ &= \cos\phi_j \frac{\partial}{\partial x'_j} - \sin\phi_j \frac{\partial}{\partial y'_j} \\ \frac{\partial}{\partial y} &= \sin\phi_j \frac{\partial}{\partial x'_j} + \cos\phi_j \frac{\partial}{\partial y'_y}, \end{aligned} \tag{30}$$

we obtain

$$L(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}) = L'(\frac{\partial}{\partial x'_j}, \frac{\partial}{\partial y'_j}), \tag{31}$$

where the coefficients in L' may depend on ϕ_j.

Since Laplace's equation is invariant under a rotation of axes, then eqn.(7) is invariant and hence transforms to

$$(\frac{\partial^2}{\partial x'^2_j} + \frac{\partial^2}{\partial y'^2_j})u = 0 \tag{32}$$

Solutions of eqn.(32) that are trigonometric in x'_j will be of the form

$$u = F(y')\sin(mx' + \alpha), \tag{33}$$

and on substituting in (32) it follows that

$$\frac{d^2F}{dy'^2_j} - m^2F = 0,$$

the solution to which is

$$F(y'_j) = Ae^{-my'_j} + Be^{my'_j}$$

On seeking functions that could represent the propogation inwards of the decaying effects of boundary adjustments on side j, we must exclude the positive exponential part. Accordingly

$$u = A_{mj}.e^{-my'_j} \sin(mx' + \alpha), \tag{33a}$$

where the arbitrary constant A may change with m and j and is written as A_{mj}. We define, as in Quinlan (1), on setting $\alpha=0$, <u>Edge-Functions for straight edges</u> in a Laplace problem:

$$E_{mj} = A_{mj}\, e^{-my'_j} \sin mx'_j \tag{34}$$

where further analysis is required to show that m must be restricted to the discrete set (36).

As in the case of V_{kj} this is a point function and can, if required, be written as $E_{mj}(P)$.

EDGE-FUNCTIONS (half-strip problem)

Recent developments in Edge-Functions show that it is useful to consider the basic problem (b) as approaching the solution in a semi-finite strip on the side AB, ($y_j' = 0$) as base, with infinite sides $x_j' = 0$ and $x_j' = a_j$. Since the other sides of the boundary are assumed to have little effect on the solution along A^*B^*, it follows that the insertion of the semi-infinite sides, with appropriate boundary conditions, will likewise have little effect on the solution along A^*B^*.

Accordingly in the case of sine series expansion (12), we set up the half-strip problem, for Fig. 2 under, with the boundary conditions

$$\begin{aligned} &u = 0 \ ; \quad x_j' = 0 \ , \quad x_j' = a_j \\ &u \to 0 \text{ as } y_j' \to \infty \\ &u = G(x_j') \ ; \quad y_j' = 0, \ 0 \le x_j' \le a_j . \end{aligned} \tag{35}$$

We seek a solution, in the usual manner, to eqn.(32) that is trigonometric in x_j' - corresponding to the pair of zero boundary conditions - in the form

$$u = F(y_j')\sin(mx_j' + \alpha)$$

where, as in (33a), on solving for $F(y_j')$ and applying condition $u \to 0$ as $y_j' \to \infty$, we obtain

$$u = A_{mj}\, e^{-my_j'} \sin(mx_j' + \alpha) \tag{35a}$$

Conditions $u = 0$ when $x_j' = 0$ and $x_j' = a_j$ require

$$\alpha = 0 \ ; \quad m = \pi M/a_j \ ; \quad M = 1,2,... \tag{36}$$

The solution for an <u>actual</u> half-strip follows, in the usual way, on taking the superposition

$$u = \sum_{M=1}^{\infty} A_{mj}\, e^{-my_j'} \sin mx_j' \ ; \quad m = \pi M/a_j \tag{37}$$

where the remaining condition (35) requires that

$$G(x_j') = \sum_{M=1}^{\infty} A_{mj} \sin mx_j' \ ;$$

the coefficients A_{mj} for a Fourier sine series expansion being given by

$$A_{mj} = \frac{2}{a_j} \int_0^{a_j} G(x_j') \sin mx_j' \, dx_j' \tag{38}$$

This analysis indicates that Edge-Functions of type (37), as defined in (34) but restricted to the discrete values $m = \pi M/a_j$, should be in solution "mix" (8). However the coefficients A_{mj} must be obtained by some scheme, as in (9), that allows for the effects of the other boundaries - rather than A_{mj} as given by (38) using the crude mathematical half-strip model. It might reasonably be expected that, when $G(x_j')$ is not identically zero, the actual calculated coefficients would be of the same order as these resulting from (38).

If expansion (12) is truncated at L harmonics, it follows that the series (37) should be truncated similarly at M = L, or in solution "mix" (8), we should include as many Edge-Functions from each side as we have significant - i.e. retained - harmonics on that side.

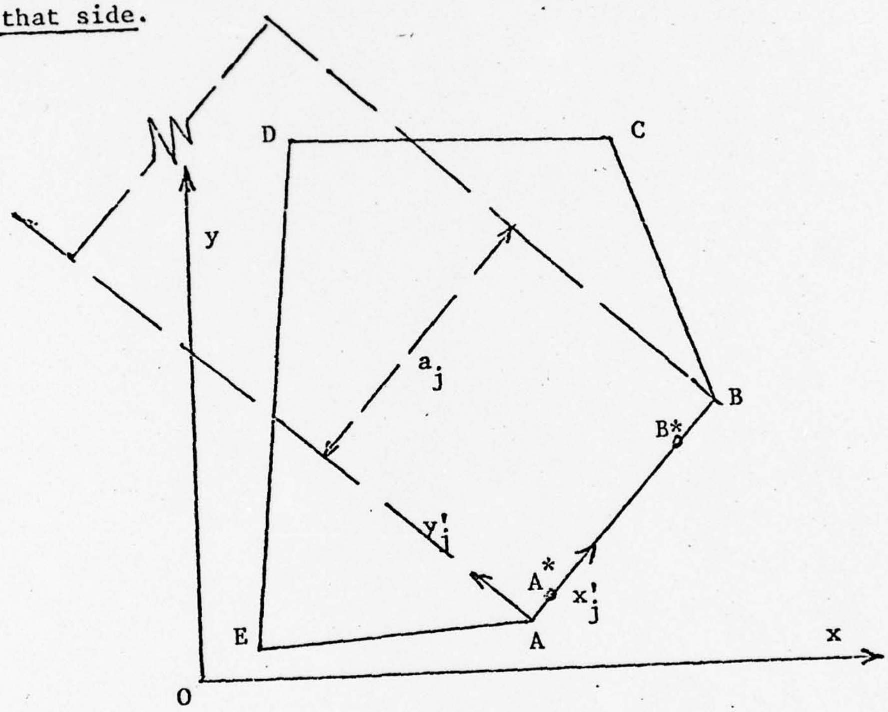


Fig. 2

It is evident that there is a distinct advantage in using the above half-strip model in place of the half-plane model in 20(b).

It remains to provide additional functions equal to the total number of vertex eqs.(19) in the problem - and the obvious sources to provide these are the Vertex Functions V_{kj}. It has been found in practice that it does not matter how many are taken from each of the various vertices, provided that sufficient of the lower eigenvalues $\lambda_1, \lambda_2, \lambda_3 \ldots$ are included from each vertex to imbed its asymptotic behaviour into the solution form.

EXAMPLE

It is a simple programming task to set up matrix (17), using eqs.(16) and (19) with truncation at any specified level L, for a polygonal region with continuous temperatures on the boundary. A simple program, LAPEX, is given in Appendix A(1), with some explanatory captions.

Appendix A(2) gives results for a quadilateral region with zero temperatures on three edges, while on the remaining edge q = 1 condition (2) is specified by

$$u = g_1(x,y) = t(1 - t) \; ; \quad t = x_1'/a_1 \tag{37}$$

The integrals in eqs.(16) are evaluated by the trapezoidal rule, noting that the integrand is zero at both ends of the range since sin $\eta x_q'$ is then zero. This anticipates the replacement of Harmonic Matching by the much more economical Harmonic Fitting Method as given in Section 5.

The equations are grouped in successive harmonic sets N = 1, N = 2....... N = L, the equations from each side being arranged consecutively in each set. The columns for the Edge-Functions and the rows for the corresponding harmonic equations are arranged to intersect on the diagonal of the coefficients matrix. This produces diagonally dominant matrix for the E.F. coefficients - the characteristic matrix form for E.F.M. The four vertex equations are arranged in rows after the harmonic equations, and the columns for the corresponding vertex functions follow those for the edge-functions. The right hand side is put up as the final column of the matrix, and the attached simple solution subroutine, Qsolve, solves the matrix using Gaussian elimination.

With a view to their extension in Appendix B simple subroutines EDGEF and POLW are introduced for the evaluation of the respective Edge and Vertex Functions at any point (xk,yk).

The Boundary Residuals - the difference between the values computed from the solution form (8) as obtained and the prescribed boundary values - are computed and printed at a specified MDIV points on each side. Their root mean square is obtained and printed as a suitable overall measure of how well the boundary conditions are satisfied on each side.

A single data card specifying points $A(u_1,v_1)$ and $B(u_2,v_2)$ and MDIV, instructs the program to compute the temperature at each of the MDIV equidistant points on the line AB. In the production phase use is made of that part of the program set up for the calculation of the boundary residuals, merely by cutting the j-cycle to a single pass, corresponding to j = 1. An indicator N code = 1 indicates the production phase; the boundary residuals phase being denoted by Ncode = 0. A blank card, which reads MDIV as zero, terminates the program.

SECTION 2

Mixed Boundary Conditions and Full Harmonic Matching

We now proceed to generalise to mixed boundary conditions, where either u or $\frac{\partial u}{\partial y'_q}$ can be specified on any edge segment, for regions with or without holes. If the boundary identity (11) for u is expanded in a full Fourier series in the range $(0,a_q)$, eqn.(12) is replaced by the series

$$\psi_q = \sum_{N=0}^{\infty} B_N \cos nx'_q + \sum_{N=1}^{\infty} C_N \sin nx'_q \equiv 0 \; ; \; n = 2\pi N/a_q \quad . \tag{39}$$

Truncation at L harmonics gives, on zeroing the relevant harmonics,

$$B_N = 0 \; ; \quad C_N = 0 \; ; \quad N = 1 \text{ to } L \tag{40}$$

$$B_0 = 0 \; ; \quad \text{(zero harmonic term)}$$

This process will be called Full Harmonic Matching, and the corresponding 2L + 1 eqs., analogous to eqs.(16), can be written as

$$\sum_{i=1}^{I} A_i \int_0^{a_q} u_i^c(x'_q) \cos(nx'_q + \alpha_k)dx'_q - \int_0^{a_q} g_q(x'_q)\cos(nx'_q+\alpha_k)dx'_q=0, \quad (4$$

for the sequence N = 0 ; N = 1 with k = 1 and k = 2,........ ....; N = L with k = 1 and k = 2; where

$$\alpha_1 = 0 \; ; \quad \alpha_2 = \pi/2 \quad ($$

Each side, including each hole, contributes its set of harmonic equations, and the corresponding cosine and sine harmonics from all sides are grouped together in harmonic sets N = 0,1.....L, to constitute the rows of the coefficients matrix.

Again as in (18) convergence of series (39) to the order $1/N^3$ can be insured by setting, in accordance with Fourier series theory,

$$\left.\begin{aligned} \psi_q(x'_q) &= 0 \\ \frac{\partial\psi_q}{\partial x'_q}(x'_q) &= 0 \end{aligned}\right\} \quad x'_q = 0 \text{ and } x'_q = a_q \tag{43}$$

A similar analysis applies on any segment q on which the normal derivative $\frac{\partial u}{\partial y'_q}$ is prescribed, in which u is replaced by $\frac{\partial u}{\partial y'_q}$ in the boundary identity (10) and $\psi_q(x'_q)$ is defined accordingly.

Four vertex equations follow from eqs.(43) for each boundary segment, though care must be taken, as in subroutine COLMAT in Appendix B, to omit equations that are redundant - as is the case when u is continuous at a vertex.

The computation of eqs.(43) for all possible boundary conditions requires that the following derivatives:

$$\frac{\partial}{\partial y'_q} \; ; \; \frac{\partial}{\partial x'_q} \; ; \; \frac{\partial^2}{\partial x'_q \, \partial y'_q} \, , \tag{44}$$

be available for all functions in solution (8).

We now proceed to reexamine the basic problems, and to obtain the necessary derivatives.

[a] Vertex Functions for the j^{th} vertex

When mixed boundary conditions occur on the sides of sector in Fig.3, we identify four possibilities, distinguished by the indicator NVEX = NVER(j), where

(i) Case I: NVER(j) = 1 corresponding to conditions u = 0 on both sides j and (j-1) of sector, which as in (25) gives

$$A = 0 \; ; \; B = 1 \; ; \; \lambda_k = k\pi/\alpha \; ; \; k = 1,2,..... \tag{45}$$

(ii) Case II: NVER(j) = 2 corresponding to conditions u = 0 on side j on which θ = 0, and $\frac{\partial u}{\partial n} = 0$ on side j-1 on which $\theta = \alpha_j$; $\hat{n}$ denotes the normal to side θ = α and can be denoted by the unit vector $\hat{\theta}$.

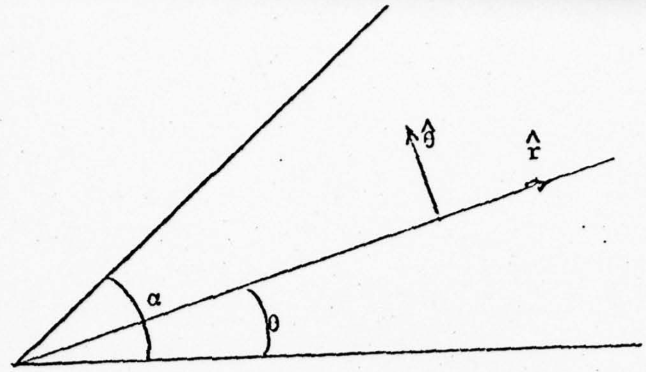


Fig. 3

Since in polar coordinates

$$\frac{\partial u}{\partial n} = \hat{\theta}.\nabla u = \hat{\theta}.[\frac{\partial u}{\partial r}\hat{r} + \frac{1}{r}\frac{\partial u}{\partial \theta}\hat{\theta}] \tag{46}$$

it follows on applying to eqn.(21) that

$$\frac{\partial u}{\partial n} = \lambda r^{\lambda-1}[-A \sin\lambda\theta + B \cos\lambda\theta] \tag{47}$$

Accordingly on analysing the sector conditions, as in (22), we obtain

$$A = 0 \;;\quad B = 1 \;;\quad \cos\lambda\alpha = 0, \tag{48}$$

the eigenvalues of which are

$$\lambda_k = k\pi/\alpha - \pi/2\alpha \;;\quad k = 1,2,.... \tag{49}$$

and the corresponding corner functions are given by form (21) on using the above values for A,B, and λ.

(iii) Case III: NVER(j) = 3 corresponding to conditions $\frac{\partial u}{\partial n} = 0$ on $\theta = 0$, and $u = 0$ on $\theta = \alpha$. Results follow similarly and are given in table I under.

(iv) Case IV: NVER(j) = 4 corresponding to conditions $\frac{\partial u}{\partial n} = 0$ on $\theta = 0$, and $\frac{\partial u}{\partial n} = 0$ on $\theta = \alpha$. All results are collected together in table I under, in a form that facilitates programming.

NVEX	λ_k	A	B
1	$k\pi/\alpha$	0	1
2	$k\pi/\alpha - \pi/(2\alpha)$	0	1
3	$k\pi/\alpha - \pi/(2\alpha)$	1	0
4	$k\pi/\alpha$	1	0

TABLE I - VERTEX FUNCTIONS

The corresponding Vertex Functions for j^{th} vertex are denoted by V_{kj} where

$$V_{kj} \equiv u = r^{\lambda_k}[A \cos\lambda_k\theta + B \sin\lambda_k\theta]; \tag{50}$$

The physics of the problem requires that λ_k be positive. The normal derivative for any direction $\hat{y}'_q$ follows, as in (46):

$$\frac{\partial u}{\partial y'_q} = \hat{y}'_q \cdot \nabla u = \hat{y}'_q \cdot [\frac{\partial u}{\partial r}\hat{r} + \frac{1}{r}\frac{\partial u}{\partial \theta}\hat{\theta}]$$

$$= \{-A\sin(\lambda'\theta + \phi_{qj}) + B\cos(\lambda'\theta + \phi_{qj})\}\lambda r^{\lambda'}; \; \lambda' = \lambda - 1 \tag{51}$$

$$\phi_{qj} = \phi_q - \phi_j$$

on noting that

$$\hat{y}'_q \cdot \hat{r} = \cos(\pi/2 + \phi_{qj} - \theta) = -\sin(\phi_{qj} - \theta)$$

$$\hat{y}'_q \cdot \hat{\theta} = \cos(\phi_{qj} - \theta)$$

where $\hat{y}'_q$ is a unit vector, normal to direction $\hat{x}'_q$, which makes an angle ϕ_q with y axis, or an angle $\pi/2 + \phi_{qj}$ with the jth side.

[b] <u>(Full) Edge-Functions</u>

Since expansion (39) involves a full Fourier Series expansion on the side AB in Fig.1 with periodic extension to all other points on the infinite line of which AB is part, it is appropriate to introduce periodicity into boundary conditions (35) for the half-strip problem of Fig.2. Accordingly we take the boundary conditions on the infinite sides of the strip as

$$u(0,y'_j) = u(a_j,y'_j) \;;\quad y'_j \geq 0, \tag{52}$$

which on applying to solution (35a) require

$$m = 2\pi M/a_j \tag{53}$$

Superposition (37) must then be replaced by

$$u = A_0 + \sum_{M=1}^{\infty} A_{mj}e^{-my'_j}\cos mx'_j + \sum_{M=1}^{\infty} B_{mj}e^{-my'_j}\sin mx'_j, \tag{54}$$

where Fourier integrals, similar to (38), follow for A_{mj} and B_{mj} in an <u>actual</u> half-strip problem. As previously we associate the cosine and sine edge-functions in (54) with the corresponding cosine and sine harmonics, (M corresponds to N) in expansion (39) - and arrange that the corresponding columns and rows intersect on the diagonal of the coefficients matrix.

The above cosine and sine edge-functions are combined in the following definition:

(Full) Edge-Functions, for side j.

$$E_{mj}=E^1_{mj}+E^2_{mj} \;;\quad E^k_{mj} = A^k_{mj}\, e^{-m_j y'_j} \cos(m_j x'_j + \alpha_k);\; k = 1,2$$

$$m_j = 2\pi M/a_j \;;\quad \alpha_1 = 0,\ \alpha_2 = \pi/2, \tag{55}$$

where k = 1 and k = 2 give the respective cosine and sine edge-functions. The term (Full) is usually omitted, as expansion (39) is more economical computerwise than the sine series expansion in (12).

If truncation on any side j in expansion (39) is taken at L, then L (Full) Edge-functions from side j should be included in solution "mix" (8) to match up the corresponding harmonics for N = 1,2,.....L. This leaves the zero harmonics in (41) - the total number for the problem being tallied as NZERO - unmatched by corresponding edge-functions, and an additional NZERO Vertex Functions must be provided to match these. The equations for the zero harmonics are inserted, as rows in the matrix, directly after harmonic set L.

As in eqn.(51) the required normal derivative for the direction $\hat{y}'_q$ follows as:

$$\begin{aligned}\frac{\partial E^k_{mj}}{\partial y'_q} &= \hat{y}'_q \cdot \nabla(E^k_{mj}) \\ &= \hat{y}'_q \cdot \{\hat{x}'_j \frac{\partial E^k_{mj}}{\partial x'_j} + \hat{y}'_j \frac{\partial E^k_{mj}}{\partial y'_j}\} \\ &= -A^k_{mj}\, m_j\, e^{-m_j y'_j} \cos(m_j x'_j + \phi_{qj} + \alpha_k);\ \phi_{qj}=\phi_q-\phi_j,\end{aligned} \tag{56}$$

since

$$\hat{y}'_q \cdot \hat{x}'_j = -\sin\phi_{qj} \;;\quad \hat{y}'_q \cdot \hat{y}'_j = \cos\phi_{qj}.$$

[c] Curved Boundaries Including Holes

The polar solutions of Laplace's eqn. in (21) apply to closed curves - where θ acts as a parameter for points in the boundary with a range $(0,2\pi)$ - provided that λ is an integer, say $\lambda = k$. Positive values for k correspond to an outer boundary, and negative values to an inner one.

Development follows as above for (Full) Edge-Functions, where the range of 2π for θ corresponds to a_j for x'_j. Accordingly when truncation is at L, for any closed curve we must include in set(8) the functions

$$\sum_{j=1}^{j'}{}^{*} \{ \sum_{k=0}^{L} P_{kj} \} \tag{57}$$

where the asterisk denotes that the terms only apply to closed curved boundaries j arising in the problem, with

$$P_{kj} = A_{Mj}\, r^{M_j} \cos(M_j\theta) + B_{Mj}\, r^{M_j} \sin(M_j\theta)$$

$$M = 1,2,3,\ldots\ldots\ldots$$

$$\begin{aligned} M_j &= M, \text{ if j relates to a curved outer boundary} \\ &= -M, \text{ if j relates to a curved inner boundary} \end{aligned} \tag{58}$$

When k = 0, we require the zero order harmonics

$$P_{0j} = A_{0j} + B_{0j} \log r \tag{59}$$

where the log r term is omitted if the section is solid. The origin for the coordinates (r,θ) can be taken at any suitable point within the corresponding closed curve, and could be taken at different locations if both inner and outer boundaries are closed curves.

The normal derivative $\frac{\partial P_{kj}}{\partial y'_q}$ follows from eqn.(51) on putting $\lambda = N_k$.

Note: Functions (58) may be thought of as Edge-Functions for the corresponding curved boundaries, and must be inserted in the appropriate rows in the harmonic sets in the Matrix.

[d] Logarithmic Singularities at Vertices: Log-Vertex Functions

As noted in the analysis of vertex functions (21) infinite derivatives of order p, where $p > \lambda_k$, occur when λ_k is not an integer. On performing a limiting analysis as $\lambda_k \to K^*$, where K^* is an integer, it can be shown that the solution to eqn.(20) denoted by V^*_{kj}, for $\lambda = \lambda_k = K^*$, where

$$V^*_{kj} = \frac{\partial}{\partial\lambda}[Ar^{\lambda}\cos\lambda\theta + Br^{\lambda}\sin\lambda\theta], \tag{60}$$

$$\begin{aligned} &= [Ar^{\lambda}\cos\lambda\theta + Br^{\lambda}\sin\lambda\theta]\log r \\ &\quad -A\lambda r^{\lambda}\sin\lambda\theta + B\lambda r^{\lambda}\cos\lambda\theta, \end{aligned} \tag{61}$$

satisfies the following conditions:-

(i) Function (60) satisfies Laplace eqn.(20) since the derivative w.r.t. the parameter λ of any solution (21) of eqn.(20) also satisfies the same equation.

(ii) the coefficient of log r - the singular part of the solution - is zero on the sides $\theta = 0$ and $\theta = \alpha$, since A, B and λ are determined in Table I for the specified boundary conditions on the angle. However, in the interior of the angle, $0 < \theta < \alpha$, the coefficient of log r is not zero, and hence the solution provides logarithmic singular behaviour within the angle.

(iii) an analysis, as in (27), of the total heat flow from term V^*_{kj} through any circular arc isolating the vertex, gives - on interchanging the operations of differentiation and integration -

$$\begin{aligned} q_j &= \frac{\partial}{\partial\lambda}\left[r^{\lambda}(1 - \cos\lambda\alpha)\right] \\ &= r^{\lambda}\left[(1 - \cos\lambda\alpha)\log r + \lambda\sin\lambda\alpha\right],\ \lambda = \lambda_k\ ; \end{aligned} \tag{62}$$

and consequently form (60) is physically admissable in solution "mix"(8) provided condition (28), $\lambda_k > 0$, is satisfied.

Note: Functions V^*_{kj} will subsequently be called LOG-VERTEX FUNCTIONS.

[e] Derivatives required in set (44)

The derivative $\frac{\partial}{\partial x'_q}$ follows from those given in eqs.(51) and (56) on replacing ϕ_{qj} by $\phi_{qj} - \pi/2$, since the direction $\hat{x}'_q$ is $\pi/2$ behind that of $\hat{y}'_q$.

On applying the operator

$$\frac{\partial}{\partial x'_q} = \hat{x}'_q \cdot \nabla,$$

to eqs.(51) and (56) respectively we obtain

$$\frac{\partial^2 V_{kj}}{\partial x'_q \partial y'_q} = \{-A\sin(\lambda''\theta + 2\phi_{qj}) + B\cos(\lambda''\theta + 2\phi_{qj})\}\lambda\,\lambda' r^{\lambda''},$$

$$\lambda'' = \lambda - 2;\ \lambda' = \lambda - 1$$

$$\frac{\partial^2 E_{kj}}{\partial x'_q \partial y'_q} = A^k_{mj}\, m_j^2\, e^{-m_j y'_j} \sin(m_j x'_j + 2\phi_{qj} + \alpha_k) \tag{63}$$

Derivatives of functions (58) follow from these in (63) on putting $\lambda = N_k$.

Difficulties are encountered in deriving analytical expressions for derivatives for Vertex Eqs.(43) in the case of curved boundary segments, as shown in section 3(a), and numerical differentiation is recommended in these cases.

Accordingly to enable the program LAPGEN in Appendix 2 to deal with all boundary segments, whether straight or curved, numerical differentiation is used for derivatives $\frac{\partial \psi_q}{\partial x_q'}$ required in Vertex Eqs.(45) for all straight line segments. This is based on the two-point formula

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}, \tag{64}$$

and the corresponding computation - involving the computation of two point values - we shall refer to as a "two-point" equation. Similarly analytical expressions for the derivatives of V^*_{kj} in (60) are best obtained by computing numerically, using formula (64), the $\frac{\partial}{\partial \lambda}$ of the corresponding derivatives of V_{kj}. These are given in POLW, where the increment $\Delta\lambda$ = delta is read in.

PROGRAM LAPGEN

The program in Appendix A(1) now requires to be expanded to include:

(i) Examine all vertex eqs. for redundancy, and - for convenience - set up all information to enable the corresponding point-eqs. to be put up in rows of matrix, after those for the zero harmonics. The assigned subscripted variables are - position, (XFIX,YFIX); associated slope, BFIX; side, JFIX; function indicator MBFIX; location subscript, (LOC); and position on side, (DFIX). The number of equations is tallied by NFP.

(ii) Total no. Vertex Functions NORF required is

$$NORF = NFP + NZERO \tag{66}$$

where NZERO is total no. zero harmonics (= no. of sides NS in a Laplace problem).

(iii) Provide for distribution of NORF Vertex Functions between the vertices of polygon, the number required at j^{th} vertex being computed as NAD(j). Locate any integer values for λ_k and arrange to include the corresponding Log-Vertex functions (60).

(iv) Set up, in subscripted locations, the information required for computing the corresponding column sets, NP, of the matrix:- function type NFN; vertex location NA; associated A and B from Table 1 in coef(1,NP), coef(2,NP); and eigenvalue λ_k in E. It is economical to put up functions (52) and (57) two columns at the time - corresponding to each value of M - and each two such columns is said to constitute a column set. Vertex functions (51) are put up in single columns, or column sets of one.

(v) Put up right hand side of matrix as its last column set, LP.

The above tasks are performed in subroutine COLMAT in Appendix B(1). Function indicators, MT, are introduced, in accordance with the following table:

MT	1	2	11	12	3
Function	u	$\frac{\partial u}{\partial y'_q}$	$\frac{\partial u}{\partial x'_q}$	$\frac{\partial^2 u}{\partial x'_q \, \partial y'_q}$	v

Table 2: Function Indicators

Note The harmonic conjugate v of u is given by MT = 3.

A more elaborate solution program SOLCOR is required, which uses Gaussian elimination on the harmonic eqs., and single pivoting to locate the maximum element in the corresponding column for zero harmonic and point-eqs. These latter equations, since they are matched by Vertex functions, cannot be put up in a diagonally dominant pattern in the matrix.

SOLCOR solves coefficients eqs.(17) not only for the specified truncation level at L harmonics, but can also obtain,as a bye product - requiring only a few percent increase in the overall computer time - the solutions for respective truncations at L-1, L-2 and L-3 harmonics.

The indicator LS specifies the number of solutions required where LS $\leq$ 4 in the present program. Accordingly, exact solutions are obtained to LS mathematical models, of increasing complexity, for the given problem.

Provision has to be made to provide the derivatives listed in set (44) for the point-fns. computed in subroutines EDGEF and POLW. Subroutines POLC and PMAP are added to provide respective Curved Edge Functions for circular or elliptic boundary segments and/or corresponding shaped holes.

Consequential adjustments are required in LAPEX in the information input, and in the putting up of the matrix. Points on ellipses are located, if required, by the usual parametric coordinates $x = a \cos \phi$, $y = b \sin \phi$, and are computed, together with corresponding slopes, in subroutine ELPS.

Provision is made to adjust the number of integration points, NOK(j), on any side by reading in a proportionality factor NHS(j). These are roughly proportional to the lengths of the corresponding boundary segments, and

$$\text{NOK(j)} = 2\text{*FSET*L*NHS(j)} + 3 \tag{67}$$

where FSET is a factor $\geq$ 1. Fset increases proportionally the number of integration points on all sides, as is required later when using Harmonic Fitting, as developed in Section 5, in place of Harmonic Matching.

An indicator NBDY is used to indicate the type of boundary conditions (2) arising in the problem. Thus NBDY = 1 indicates that on the j^{th} side - where the occurance of non-zero boundary conditions is indicated by the subscripted variable NBY(j) = 1 - the boundary conditions are of the polynomial form

$$g_j(x,y) = \sum_{k=1}^{6} C(j,k)t^{k-1} , \qquad (65)$$

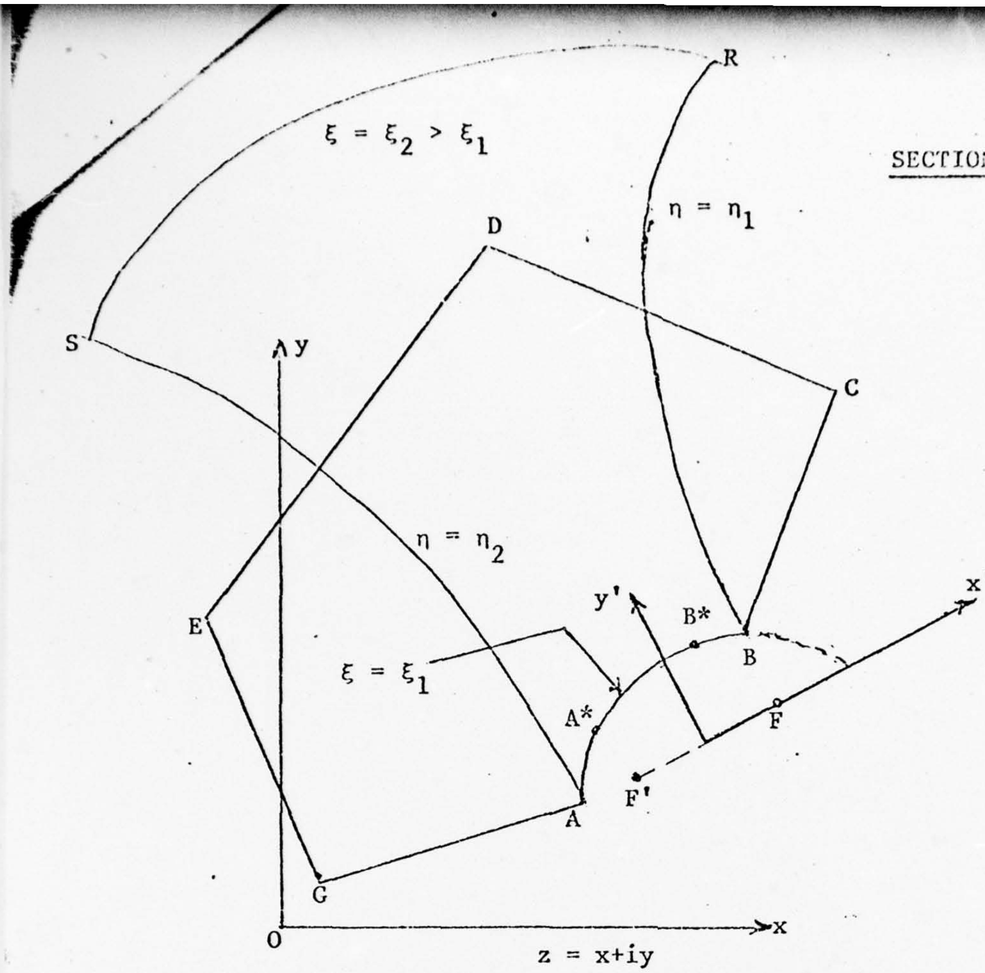
where $t = x_j'/a_j$. The required coefficients C(j,k) are read in for each side which has NBY(j) = 1.

Computation of point values of (65) is done in a subroutine POLP, which also provides numerical coefficients - without requiring to read them in as in the case of NBDY = 1 - for the torsion problem as indicated by NBDY = 2. Additional cases can be inserted, corresponding to values of NBDY > 2, in POLP as the user requires.

The checking of the boundary residuals is as before except that it is done for LS truncation levels.

Production is based on one data card for each line of production, but is now done in a special subroutine PRODN, which enables various operations indicated by NCODE = 1,2,3 - differentiations, principal stress computations and integrations - to be performed. The user can readily incorporate into PRODN any further facilities that may be required.

The expanded program, LAPGEN, with explanatory captions is attached in appendix B(1), and an illustrative example is given in Appendix B(2). A user guide to inputting data on the black box principle is given in Appendix C.



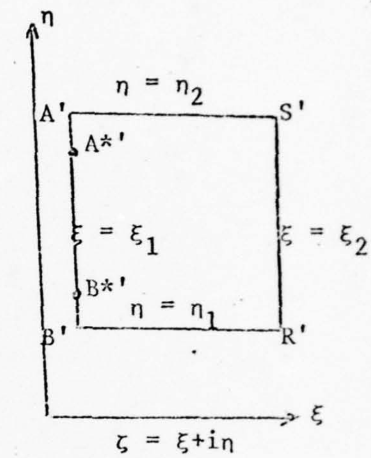


Fig. 4

(a) Curved Edge-Functions for Curved Boundary Segment AB

If the mapping

$$\zeta = F(z) \tag{68}$$

maps the rectilinear region A'B'R'S' in the ζ-plane onto the curvilinear region ABRS in the z-plane, then any solution of Laplace's eq.(7) in the ζ-plane satisfies Laplace's eq. in the z-plane.

Analogous to the half-strip approximation (35) to basic problem (b) at (20), the solution near A*B* - where the other sides are assumed to have little effect - has the characteristics of the solution in ABRS in the region A*B*, when, as in (52), we set

$$u(\xi, \eta_1) = u(\xi, \eta_2) \ ; \quad \xi \geq \xi_1, \tag{69}$$

as the conditions on sides BR and AS respectively.
These conditions can be satisfied by requiring that u be periodic in η, the period being $\eta_2 - \eta_1$. In the transformed problem in the ζ-plane, the solution in the region of A^*B^* is characterised by edge-functions, analogous to functions (55), which may be written in the form

$$C_{m_j} = C^1_{m_j}+C^2_{m_j}; \quad C^k_{m_j} = A^k_{m_j}\, e^{-m_j\xi_j} \cos(m_j\eta_j+\alpha_k); \quad k = 1,2$$

$$m_j = 2\pi M/(\eta_2-\eta_1); \quad \alpha_1 = 0, \; \alpha_2 = \pi/2 \tag{70}$$

where the mapped plane ζ corresponding to the j^{th} side, when curved, is denoted by

$$\zeta_j = \xi_j + i\eta_j, \tag{71}$$

and may be termed the ζ_j-plane.

In subsequent work it is advantageous to work with the complex variable ζ_j, and to write function (70) in the form

$$C_{m_j} = \mathrm{Re}\{A_{m_j}\, e^{-m_j\zeta_j}\} = e^{-m_j\xi_j}[A'_{m_j} \cos(m_j\eta_j)-A''_{m_j} \sin(m_j\eta_j)] \tag{72}$$

where A_{m_j} is a complex constant

$$A_{m_j} = A'_{m_j} - iA''_{m_j} \tag{73}$$

and Re denotes the real part.

Accordingly, the appropriate functions C_{m_j} must be introduced into solution mix (8) for each curved edge-segment.

We now proceed to find the derivatives of C_{m_j} as required in set (44).

Analogous to mapping (68) we can write the mapping for curved side j as

$$\zeta_j = F_j(z) \tag{74}$$

Since C_{m_j} is a function of z and $\bar{z}$, say

$$C_{m_j} = G(z,\bar{z}) \; ; \quad z = x+iy \; ; \quad \bar{z} = x-iy \; .$$

we require the operators

$$\frac{\partial}{\partial x'_q} = e^{i\phi_q}\frac{\partial}{\partial z} + e^{-i\phi_q}\frac{\partial}{\partial \bar{z}}$$

$$\frac{\partial}{\partial y'_q} = ie^{i\phi_q}\frac{\partial}{\partial z} - ie^{-i\phi_q}\frac{\partial}{\partial \bar{z}} \tag{75}$$

The operator $\frac{\partial}{\partial x'_q}$ follows on noting, from fig.1, that the coordinates of any point P(x,y) are given by:

$$x = x_q + x'_q \cos\phi_q - y'_q \sin\phi_q$$

$$y = y_q + x'_q \sin\phi_q + y' \cos\phi_q \, , \tag{76}$$

where the coordinates of the q^{th} vertex are (x_q, y_q). On operating on $G(z,\bar{z})$ it follows that

$$\frac{\partial G}{\partial x'_q} = \frac{\partial G}{\partial z}\frac{\partial z}{\partial x'_q} + \frac{\partial G}{\partial \bar{z}}\frac{\partial \bar{z}}{\partial x'_q},$$

and on substituting

$$\frac{\partial z}{\partial x'_q} = \frac{\partial x}{\partial x'_q} + i\,\frac{\partial y}{\partial x'_q} = e^{i\phi_q} \;;\quad \frac{\partial \bar{z}}{\partial x'_q} = e^{-i\phi_q},$$

the operator for $\frac{\partial}{\partial x'_q}$ is obtained. The operator $\frac{\partial}{\partial y'_q}$ follows similarly.

Accordingly we obtain

$$\frac{\partial C_{mj}}{\partial y'_q} = \mathrm{Re}[A_{mj}\frac{\partial}{\partial y'_q}(e^{-m_j \zeta_j})]$$

$$= \mathrm{Re}[-im_j\, A_{mj}\, e^{i\phi_q}\, \frac{d\zeta_j}{dz}\, e^{-m_j\zeta_j}] \tag{78}$$

The parameter η_q, for segment q, now corresponds to x'_q - the parameter for points on qth line segment - as used in harmonic matching eqs.(41). Analogous to the vertex eqs.(43), we require derivatives w.r.t. η_q for all functions in solution "mix" and for all their normal derivatives. These are difficult to compute and are best obtained by numerical differentiation using "two-point" eqs. formula (64) as set up in LAPGEN.

(b) <u>Example: Elliptic indentation</u>

If AB in Fig.4 is a segment of an ellipse, with principal axes x' and y', the required mapping is

$$z' = c\cosh\zeta;\quad z' = x' + iy' \tag{79}$$

from which the point correspondences are

$$x' = c\cosh\xi\cos\eta$$
$$y' = c\sinh\xi\sin\eta \tag{80}$$

The mapping is rendered single valued by taking the limits $\xi(0,\infty)$ and $\eta(0,2\pi)$. The corresponding singular points in the z-plane are at $F'(-c,0)$, $F(c,0)$ with a cut from $-c$ to c on the x' axis.

On eliminating η from eqs(80) it follows that

$$\frac{x'^2}{c^2\cosh^2\xi} + \frac{y'^2}{c^2\sinh^2\xi} = 1 \ , \tag{81}$$

or any line $\xi = \xi_0$ in the ζ-plane corresponds to an ellipse in the z'-plane with semi-axes $a = c \cosh \xi_0$, $b = \sinh \xi_0$; $c^2 = a^2 - b^2$. Similarly $\eta = \eta_0$ corresponds to a branch of a hyperbola that has the same focii as the ellipse. Then, it is easily shown that ABRS is mapped onto A'B'R'S' in a one-to-one manner.

To obtain point-values for the coefficients of A'_{mj} and A''_{mj} from C_{mj}, or its derivatives, we require to find ξ and η for any point P(x',y') in the domain Ω - where (x',y') are coordinates of P w.r.t. principal axes for elliptic indentation. Equation (81) gives the quadratic

$$c^2T^2 + T(c^2-x'^2-y'^2)-y'^2 = 0 \ ; \quad T = \sinh^2\xi \ ,$$

from which

$$2c^2T = -(c^2-x'^2-y'^2) + \sqrt{(c^2-x'^2-y'^2)^2 + 4c^2y'^2}, \tag{82}$$

the positive square root being taken since $T > 0$. When $y' = 0$ then

$$\begin{aligned} T &= x'^2/c^2 - 1 \ ; \quad x'^2 \geq c^2 \\ &= 0 \qquad\qquad\quad ; \quad x'^2 < c^2 \end{aligned} \tag{82a}$$

It follows that

$$\xi \doteq \sinh^{-1}(\sqrt{T}) = \log_e(\sqrt{T} + \sqrt{1+T}), \tag{83}$$

and eqs (80) then give

$$\eta = \tan^{-1}\left(\frac{y'\cosh\xi}{x'\sinh\xi}\right) \ , \tag{84}$$

where η must be adjusted, for quadrant location of P, to lie in $(0,2\pi)$

A suitable subroutine, PMAP, similar to EDGEF and POLC in Appendix B, is included. The values for η_1 and η_2 are obtained from (84). Note that singular points of mappings (74) are not allowable in the domain of solution (8), which must be regular at all points other than the singular points specified as in (3). If the singular points of any mapping occur in the domain Ω of the problem then either (i) split Ω into elements to ensure that the domain of any set of curved edge-functions does not include the corresponding singular points of the transformation, or (ii) superpose other suitable functions, as over in (e), to neutralise any unwanted singularities.

(c) Circular Indentations

This follows as in (b) on using the transformation

$$z = e^{\zeta} \tag{85}$$

Alternatively, if AB in Fig.4 is an arc of a circle, the corresponding basic problem can be taken as that for a curvilinear region bounded by radii $r = r_1$, $r = r_2$; and radial lines $\theta = \theta_1$, $\theta = \theta_2$, where $r_2 \to \infty$ The solution then follows in form (57) on setting

$$M_j = -2\pi M/(\theta_2 - \theta_1) \tag{86}$$

(d) Smooth Inner Boundaries

If the inner boundary is a smooth curve the corresponding Curved Edge Functions, analogous to polar functions (58), follow from eq.(82), on adding the periodicity requirement

$$\eta_1 = 0 \;; \quad \eta_2 = 2\pi \tag{87}$$

(e) Solid Elliptical Section

Let us examine the mapping

$$z = c \cosh \zeta, \tag{88}$$

as in (b), in the case of a region bounded by a solid ellipse, shown in Fig. 5.

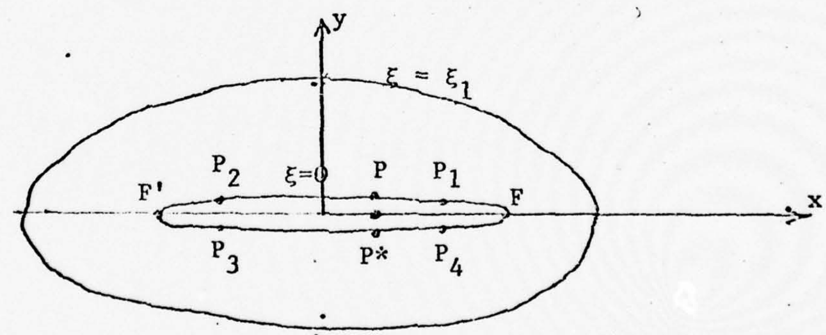


Fig. 5

The corresponding Curved Edge-Functions for $\xi = \xi_1$, analogous to those in (72), that decay inwards from the boundary, are

$$C_M = \mathrm{Re}\{A_M e^{M\zeta}\} = e^{M\xi}[A_M' \cos M\eta - A_M'' \sin M\eta] \tag{89}$$

These require a cut FF' in the z-plane, which corresponds to the limiting ellipse $\xi = 0$. On taking $\xi(0,\infty)$, $\eta(0,2\pi)$ to render mapping single-valued η is discontinuous on the cut, having values, at respective neighbouring points, (i) of $\eta = \alpha$ and $\eta = 2\pi-\alpha$ at P_1 and P_4 and (ii) of $\eta = \pi-\alpha$ and $\eta = \pi+\alpha$ at P_2 and P_3 where $0 \leq \alpha \leq \pi/2$.

It follows that cos $M\eta$ is continuous on cut FF' but sin $M\eta$ is discontinuous as it changes sign - and accordingly C_M is discontinuous. The functions got on replacing M by -M in (89)

$$C_{-M} = \mathrm{Re}\{A_{-M}\, e^{-M\zeta}\} = e^{-M\xi}[A'_{-M} \cos M\eta + A''_{-M} \sin M\eta], \tag{90}$$

have a similar discontinuity on FF', and on superposing, the resulting discontinuity on $\xi = 0$ is

$$2(A''_M - A''_{-M}) \sin M\alpha,$$

and this reduces to zero on setting

$$A''_{-M} = A''_M \tag{91}$$

Analogous to eq.(78), and noting from eq.(88) that

$$\frac{d\zeta}{dz} = \frac{1}{c \sinh \zeta}, \tag{92}$$

the discontinuity between points P and P^* in the derivation $\frac{\partial}{\partial y'_q}$ of $C_M + C_{-M}$ is, on $\xi = 0$:

$$\mathrm{Re}[iMe^{i\phi_q} \cdot \left(\frac{1}{ic \sin \eta}\right) \{A_M\, e^{iM\eta} - A_{-M}\, e^{-iM\eta}\}]_P^{P^*}$$

$$= \frac{2M}{c}\, \mathrm{Re}[e^{i\phi_q}\, (A_M - A_{-M}) \frac{\cos M\eta}{\sin \eta} + (A_M + A_{-M}) \frac{\sin M\eta}{\sin \eta}]_P^{P^*} \tag{93}$$

and hence the discontinuity in the term cos $M\eta$/sin η vanishes if

$$A_M = A_{-M} \tag{94}$$

Also <u>note</u> that at the singular points of mapping (88) the function $C_M + C_{-M}$ and its derivative $\frac{\partial}{\partial y'_q}$ is everywhere finite since limit of (sin $M\eta$/sin η) $\to$ M at points F', $\eta = \pi$, and F, $\eta = 0$ or 2π.

Similarly all the higher derivatives of $C_M + C_{-M}$ can be shown to be finite and continuous at all points within and on the elliptical boundary.

If a region is bounded internally and externally by ellipses, then the curved Edge Functions for the outer bounded must be rendered finite and continuous by superposing the corresponding C_{-M} when the focii F and F' of the outer ellipse are in the region of the problem.

(f) Approximate Curved Edge-Functions

If the mapping (79) is used to map* the side A'B of the rectangle A'B'R'S' onto an elliptic arc in the z-plane that is not too different from the curved boundary segment AB, then the functions C_{mj} as got from the elliptic mapping approximates pretty well the characteristics of the solution in region A^*B^* and can be included in solution "mix" (8).

Points on the actual boundary can be expressed in terms of ξ and η, where η is in (η_1, η_2). The parameter η can be used for the actual boundary points, the corresponding ξ being where the curve η intersects the actual boundary - and η_q, for segment q, then corresponds to parameter x'_q in eqs.(51) for harmonic matching.

Accordingly on exact mapping (79) is not essential, and in most practical cases a "fitted" elliptic mapping is sufficient.

---

*In practice this would require the fitting, by a least squares criterion, of an elliptic arc - involving five parameters (axes, centre, orientation) - to the prescribed arc AB. We shall refer to the resulting mapping as a "fitted" mapping.

SECTION 4 - OTHER BASIC PROBLEMS

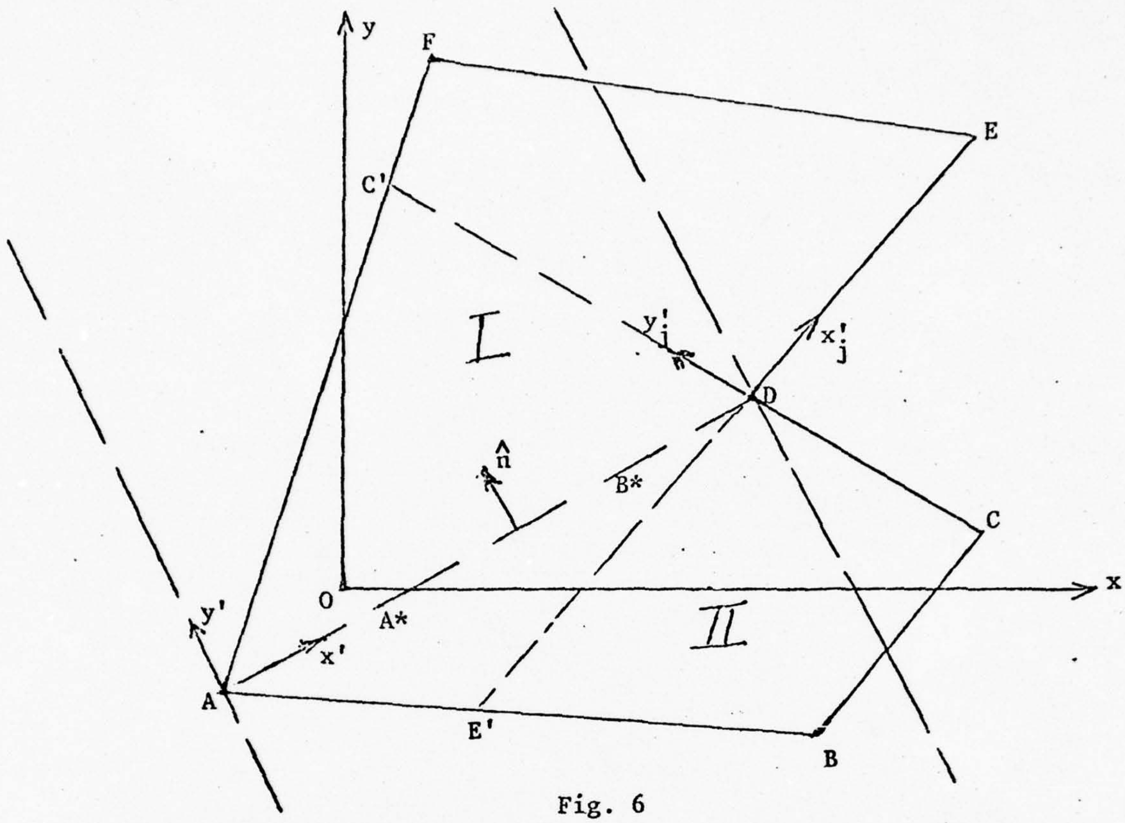


Fig. 6

(a) RE-ENTRANT ANGLE

The semi-finite, or half-strip model, used in arriving at Edge-Functions (37) or (55) does not apply to the sides of a reentrant angle, as at D in Fig.6. On considering the side DE, the assumption that the solution on DE' is related to a periodic repetition of that on DE is clearly untenable. Also in the region $y_j' < 0$ the exponential becomes positive and obviously cannot represent a propogation into the interior of the boundary actions on DE.

However if we divide up Fig. 6 into two convex regions, I and II, by taking a cut from D anywhere within the angel E'D C', say along DA, this will ensure that $y_j' > 0$ for each region, and that all the corresponding Edge-Functions have the characteristic negative exponentials at all points in their respective regions.

The solution "mix" for eq.(1) can then be written as

$$\begin{aligned} u &= u_o + u_{(i)} \\ &= u_o + u_{(ii)} \end{aligned} \tag{93}$$

for regions I and II respectively, where u_o denotes all the functions that are common to both regions and are continuous, and with continuous derivatives, across the cut.

The functions $u_{(i)}$ and $u_{(ii)}$, which are confined to regions I and II respectively, must then be matched across the cut to give continuity, in a least squares sense, across the cut for u and $\frac{\partial u}{\partial n}$.

Since the Edge-Functions from DC and DE produce a discontinuity across AD, this must be negatived by superposing functions that are characteristic of a line, AB, of discontinuities in u and $\frac{\partial u}{\partial n}$. Analogous to the half-strip model (52) the solution characteristics in region A*D* are similar to those for an infinite strip bounded by infinite sides perpendicular to AD through A and D respectively.

On taking edge-axes x' and y' for AD as shown in Fig.6, we require that the solution u(x',y') satisfy the following boundary conditions for the strip:

(i) $u(x',y') \to 0$ as $y' \to \infty$ and $y' \to -\infty$,

(ii) $u(0,y') = u(a',y')$, for all points on $x'= 0$ and $x' = a'$, where $a' = AD$.

Since u(x',y') satisfies Laplace's eqn. suitable forms for u, in semi-finite strips $y' > 0$ and $y' < 0$, analogous to forms (54), are, with $m = 2\pi M/a'$:

$$\begin{aligned} u &= \Sigma\, e^{-my'}\{A_m \cos mx' + B_m \sin mx'\} \quad , \quad y' > 0 \\ &\ \Sigma\, e^{my'}\{A'_m \cos mx' + B'_m \sin mx'\} \quad , \quad y' < 0 \end{aligned} \tag{94}$$

In an actual strip problem, if the discontinuities in u and $\frac{\partial u}{\partial y}$, were specified as f(x') and g(x') respectively, it follows that

$$f(x') = \Sigma\{(A_m - A'_m)\cos mx' + (B_m - B'_m)\sin mx'\}$$

$$g(x') = \Sigma\{-m(A_m + A'_m)\cos mx' + -m(B_m + B'_m)\sin mx'\},$$

and on using Fourier coefficients formulae the coefficients in (94) follow.

However in an actual problem, as in Fig. 6, the coefficients must be determined as in (11) which allows for the influence of all sides, and functions (94) must be included in solution "mix" (8).

These latter can be identified as Edge-Functions of type (55) for the upper and lower side of the cut AD, where the upper side has indicator j and the lower has (j + 1).

(b) SINGULARITY ON EDGE

Consider a singularity, as in (3),

$$u = Q\,\delta(\eta-\eta_o)$$

acting at the point P, parameter η_o, on curved boundary segment AB in Fig. 4. On taking ABRS as the basic region with conditions (69), and superposing solutions (72), to satisfy the conditions on AB, $\xi = \xi_1$, we require

$$Q\,\delta(\eta-\eta_o) = \Sigma\, e^{-m\xi_1}\,[A_{mj}^{(1)}\cos(m\eta) + A_{mj}^{(2)}\sin(m\eta)]. \tag{95}$$

The coefficients follow, on taking a summation M = 0 to M = ∞, as

$$A_{mj}^{(1)}\, e^{-m\xi_1} = \frac{Q}{\eta_2-\eta_1}\int_{\eta_1}^{\eta_2} \delta(\eta-\eta_o)\cos(m\eta)\,d\eta,$$

which gives for $A_{mj}^{(1)}$ and, similarly, for $A_{mj}^{(2)}$

$$A_{mj}^{(1)} = \frac{Q\, e^{m\xi_1}\cos(m\eta_o)}{\eta_2-\eta_1} \tag{96}$$

$$A_{mj}^{(2)} = \frac{Q\, e^{m\xi_1}\sin(m\eta_o)}{\eta_2-\eta_1}$$

Series (95) provides a useful <u>particular integral</u> (4) for above type of boundary singularity. Since its value is known, $Q\delta(\eta-\eta_o)$, on AB, the series only requires to be evaluated at interior points (ξ,η) where $\xi > \xi_1$, and it is then dominated by the negative exponential factor $e^{-m(\xi-\xi_1)}$.

Note (i) that all quantities $\xi_1, \eta_1, \eta_2, \eta_o$ and m relate to side j, but subscript j is omitted here, as not necessary.

(ii) If the singularity acts on a straight side we require the obvious replacements

$$\eta \Rightarrow x_q'; \quad \xi_1 = 0; \quad \eta_1 = 0; \quad \eta_2 = a_1; \quad \eta_o = a,$$

the singularity being at the point $x_q' = a$.

SECTION 5 - DISCRETE ORTHOGONAL FUNCTIONS

If the function $y=f(x)$ is given as the set of n discrete points x_λ in the interval (a,b), by

$$y_\lambda = f(x_\lambda) \; ; \quad \lambda = 1,n \tag{97}$$

and $\{\emptyset_k(x)\}$ is a set of m discrete orthogonal functions, then we can approximate f(x) by the polynomial

$$\bar{y} = \sum_{\lambda=1}^{m} c_k \emptyset_k(x) \tag{98}$$

by minimising the sum of the squares, η^2, of the weighted errors - weight factors w_λ - at the n data points x_λ:

$$\eta^2 = \sum_{\lambda=1}^{n} w_\lambda (y_\lambda - \bar{y}_\lambda)^2$$

$$= \sum_{\lambda=1}^{n} w_\lambda [y_\lambda - \sum_{k=1}^{m} c_k \emptyset_k(x_\lambda)]^2 \tag{99}$$

where $\bar{y}_\lambda$ is the value of $\bar{y}$ at x_λ.

On minimising η^2 with respect to the unknowns $c_j; j=1,\ldots,m$ by setting

$$\frac{\delta(\eta^2)}{\delta c_j} = 0 \; ; \quad j = 1,\ldots,m; \quad n \geq m \; ;$$

we obtain

$$\sum_{\lambda=1}^{n} w_\lambda [y_\lambda - \sum_{k=1}^{m} c_k \emptyset_k(x \;)] \; \emptyset_j(x_\lambda) = 0 \tag{100}$$

If the set $\{\emptyset_k(x)\}$ is orthogonal over the discrete points x_λ with weight factors w_λ, this means that

$$\sum_{\lambda=1}^{n} w_\lambda \emptyset_k(x_\lambda) \emptyset_j(x_\lambda) = N_j \; \delta_k^j \; ; \tag{101}$$

where N_j is the 'norm' of the function $\emptyset_j$, and δ_k^j is the Kronecker delta. Accordingly, equation (A-5) gives on replacing subscript j by k:

$$c_k = \frac{1}{N_k} \sum_{\lambda=1}^{n} w_\lambda y_\lambda \emptyset_k(x_\lambda) \tag{102}$$

If the set $\phi_k(x)$ is complex, then $\phi_j(x)$ in (101) and (102) must be replaced by its complex conjugate $\phi^*_j(x)$ since N_j must be real giving

$$\sum_{\lambda=1}^{n} w_\lambda \phi_k(x_\lambda)\phi_j^*(x_\lambda) = N_k \delta_k^j$$

$$c_k = \frac{1}{N_k} \sum_{\lambda=1}^{n} w_\lambda y_\lambda \phi_k^*(x_\lambda), \tag{103}$$

on noting that the right hand side of (98) must be real which implies that $c_k\phi_k(x)$ must be real, and that η^2 is real.

Two limiting cases for n are of special interest:

(i) If <u>$n=m$</u>, then the obvious minimum condition for η^2 is that each of the errors $(y_\lambda - \bar{y}_\lambda)$ should be zero, or the approximating curve $\bar{y}$ should pass through the set of points (x_λ, y_λ), $\lambda=1,\ldots,n$, and hence y and $\bar{y}$ would be point-matched, or collocated, at the above set of points. If a unique solution exists it is necessary that $n \geq m$.

(ii) If $n \to \infty$ and $m \to \infty$, then the discrete set x_λ becomes the continuous x in the interval (a,b), and (98) becomes an expansion in the infinite set of orthogonal functions $\{\phi_k(x)\}$. When $\phi_k(x)$ is complex, the coefficients follow from (103) as

$$c_k = \frac{1}{N_k} \int_a^b w(x)y(x)\phi_j^*(x)dx, \tag{104}$$

and the condition for orthogonality of the function $\phi_k(x)$ in (a,b) follows from (102) as

$$N_k\delta_k^j = \int_a^b w(x)\phi_k(x)\phi_j^*(x)dx, \tag{105}$$

where w(x) is the associated weight factor in (a,b).

<u>Discrete Fourier Polynomials</u>

If $\phi_k(x) = e^{ikx}$ in (98) it is easily shown that these functions satisfy the orthogonality relations

$$\int_{-\pi}^{\pi} e^{ikx}\, e^{-ijk}dx = 2\pi\delta_k^j \;;\; i = \sqrt{-1}, \tag{106}$$

the quadrature expression for which is

$$\mathop{Lt}_{n\to\infty} \sum_{\lambda=-n}^{n} w_\lambda e^{ikx_\lambda}\, e^{-ijx_\lambda} = 2\pi\delta_k^j, \tag{107}$$

where the weight factors are w_λ, the origin for x being taken at the mid-point of the interval $(-\pi,\pi)$, which is divided for convenience into 2n equal parts.

Result (107) shows that the functions e^{ikx_λ} are orthogonal over the infinite set of discrete points x_λ in $(-\pi,\pi)$ with weight factors w_λ. We now ask can some w_λ and distribution of points x_λ be found, resulting from the application of some quadrature formula to (106), which will satisfy relation (107) for finite values of n and thereby provide a method of deducing a set of discrete orthogonal polynomials from a corresponding set of continuous orthogonal functions.

On investigating the trapezoidal formula, with its associated weight factors

$$\begin{aligned} w_\lambda &= \tfrac{1}{2} \; ; \quad \lambda = \pm n \\ &= 1 \; ; \quad |\lambda| < n \end{aligned} \tag{108}$$

and taking the equidistant distribution for x

$$x_\lambda = \frac{\lambda\pi}{n} \quad ; \quad \lambda = -n(1)n,$$

we are led to examine the finite analogue, S, of series (107):

$$S = \sum_{\lambda=-n}^{n} w_\lambda \, e^{i(k-j)x_\lambda} = \sum_{\lambda=-n}^{n} w_\lambda \, e^{i\lambda\theta} , \tag{109}$$

where

$$\theta = \pi(k-j)/n.$$

On substituting into (109) for w_λ from (108) and applying the formula for summing (2n+1) terms of a geometrical progression, it follows, when $k \neq j$, that

$$S = -\tfrac{1}{2}(e^{-in\theta} + e^{in\theta}) + e^{-in\theta} \, \frac{1-e^{i(2n+1)\theta}}{1-e^{i\theta}} , \tag{110}$$

where

$$e^{in\theta} = e^{i\pi(k-j)} = \cos(k-j)\pi = \pm 1 \; ; \quad k \neq j,$$

depending on whether (k-j) is even or odd. Accordingly it follows that S = 0.

If k = j, then the series (109) reduces to

$$S = \sum_{\lambda=-n}^{n} w_\lambda = 2n, \tag{111}$$

and hence for all k and j

$$S = \sum_{\lambda=-n}^{n} w_\lambda \, e^{ikx_\lambda} \, e^{-ijx_\lambda} = 2n \, \delta_k^j , \tag{112}$$

which establishes condition (103) with $N_k = 2n$ for the functions e^{ikx_λ} when k is any integer, including zero. It follows from (103) that

$$c_k = \frac{1}{2n} \sum_{\lambda=-n}^{n} w_\lambda y_\lambda e^{-ikx_\lambda}$$

and if we take a correspondingly balanced polynomial form for (98) by writing

$$\bar{y} = \sum_{k=-m}^{m} c_k e^{ikx}, \tag{114}$$

it follows from (113) that

$$c_{-k} = c_k^*,$$

where c_k^* in the complex conjugate of c_k, and on writing

$$c_k = a_k + ib_k$$

series (114) reduces to the real series

$$\bar{y} = c_o + 2\sum_{k=1}^{m} a_k \cos kx - 2\sum_{k=1}^{m} b_k \sin kx, \tag{115}$$

where on taking real and imaginary parts of result (113), we obtain

$$2a_k = \frac{1}{n} \sum_{\lambda=-n}^{n} w_\lambda y_\lambda \cos kx_\lambda$$

$$2b_k = -\frac{1}{n} \sum_{\lambda=-n}^{n} w_\lambda y_\lambda \sin kx_\lambda \tag{116}$$

On setting $x = x' - \pi$, we easily show that (115) and (116) are invariant in x'. Hence the above formulae hold for the x' interval $(0,2\pi)$, and this can be mapped onto the interval (0,c) by setting $x' = 2\pi x'/c$, replacing kx' by $2\pi kx''/c$ Or, on dropping the dashes, this means that (114) and (116) apply to the interval (0,c) on replacing k by $2\pi k/c$.

Trigonometric Interpolation Series

Since boundary identity (39) can be approximate by the truncated series

$$\psi(x'_q) \quad \sum_{N=0}^{L} B_N \cos nx'_q + \sum_{N=1}^{L} C_N \sin nx'_q, \tag{117}$$

in the interval, $0 \leq x'_q \leq a_q$; $n = 2\pi N/a_q$, we might regard the trigonometric series on the r.h.s. as a trigonometric interpolation series for $\psi'(x'_q)$ in the specified interval. The fitting coefficients then follow by discrete least squares as in eq.(98), the coefficients being given by formulae (116).

We observe that the series (115) is similar to a Fourier series, truncated at m terms, for the interval $(-\pi,\pi)$ and that the summation formulae (116) for the Fourier coefficients $2a_k$ and $-2b_k$ are what would be obtained by evaluating the corresponding Fourier series integrals, analogous to these in (16), by the trapezoidal rule, involving the division of the interval $(-\pi,\pi)$ into 2n equal parts. Hence a working rule for the harmonic equations in sets (16) or (41):

Evaluate all Fourier integrals by the trapezoidal rule

This, in effect, substitutes series (116) for the corresponding integrals, as is required when harmonic matching is replaced by discrete least squares minimisation of the boundary residuals on each boundary segment.

SECTION 6 - DISTINCTIVE FEATURES

The distinctive features of the Edge-Function Method, as illustrated by the examples in Appendices A and B and the formulation in the present paper, are:

(a) Algebra:

The functions introduced in solution "mix" (8) may appear complicated at first sight, but an effective algorithmic method of controlling the resulting algebra - called the Computer Form Method - was developed by Quinlan [12,13]. Accordingly, Edge-Functions, Vertex Functions and any of their derived functions like normal slopes, moments or shears, can, on calling the appropriate subroutine, be obtained as readily as any trigonometric or exponential function.

(b) Programming:

A systems approach to programming, based on a main program Quinp together with 18 subroutines - each with its own definite task to perform - is given in [4]. This has been used to advantage by subsequent research workers, Tai, Nash, Dashmukh, O'Callaghan and others, to considerably simplify and shorten the programming tasks arising in extending the Edge-Function Method to problems of greater complexity, in vibrations and shallow shells. The program Quinp [4] and its accompanying program-description is still adequate, though it is proposed to issue shortly an up-dated version to include cuts, cracks and reentrant angles.

The program LAPGEN given in Appendix B(1) illustrates the chief features of QUINP. If one is familiar with LAPGEN, the various parts of QUINP should then be readily understandable.

(c) Computing-Time:

Considerable computing time is saved by using the discrete - rather than the continuous - least squares method of minimising the boundary residuals, as developed in Section 5, thus saving considerable time for setting up the coefficients matrix. The rows are arranged in the coefficients matrix so that the equations for each harmonic are grouped together in successive bands, thereby producing a strongly diagonalised system with a considerable saving in solution time. Moreover this arrangement of the matrix enables the solutions corresponding to a lesser number of harmonics to be deduced without any appreciable increase in computer time. Accordingly, as a routine, solution vectors corresponding to terminating the boundary identities at L-3, L-2, L-1 and L harmonics respectively are computed and tested for each problem.

(d) Acceptabiligy of Solution:

Each solution vector provides an "exact" solution to a problem governed by the same differential equation but with slightly different boundary conditions to those specified. Such solutions may be regarded as mathematical models of the physical problem. In each case the difference - termed the Boundary Residuals - between the boundary values as computed and the specified values, is computed and reported through its approximate root mean square value - r.m.s. - on each side of the boundary and for each boundary condition. The set of r.m.s. values provides a simple yet comprehensive reliability test to enable an engineer to decide whether, or not, to accept the results.

(e) Convergence Demonstration:

A practical "convergence" demonstration is provided by the routine provision in each problem of several solutions, corresponding to increasing harmonics and matrix size, with r.m.s. values presented for the resulting boundary residuals. These invariable decrease rapidly as the number of harmonics used increased. Likewise production results (e.g. normal slopes, shears, harmonic conjugate, etc.) as required at interior points, are always computed for a number of different harmonics and hence as in examples given in appendix B their "convergence" can be seen at a glance. No other competing system - finite element, finite differences or boundary integral - can offer this effective comparison of the effect on the results of increased computer time expenditure, without involving a very considerable increase in computer time over that which would be required for a one-shot solution based on the largest matrix size involved.

(f) No pre-computer processing of the problem is required

As can be seen from Appendix C and from the data cards at the end of each example, no pre-computer processing is required. Only the geometrical and load data and material's moduli are required, together with one control card. These do not require any knowledge of E.F.M. for their preparation. Accordingly E.F.M. can be operated completely as a "Black Box". This is in sharp contrast to the intricate element networks required in finite element and boundary integral methods. Production is based on a single data card for each production set, requiring the computation of a specified function at a specified number of equidistant points on a specified line.

REFERENCES FOR THE EDGE-FUNCTION METHOD

1. Quinlan, P. M. (1964), "The Torsion of an Irregular Polygon", Proc. Roy. Soc., Vol. 282A.

2. Quinlan, P. M. (1968), "Polygonal and Swept-Back plates with Cut-Outs and Column Supports", OAR Research Applications Conference, Washington, 1968.

3. Kelly, G. V. and Quinlan, P. M. (1967), "The Method for Boundary Value Problems", OAR Research Report.

4. Quinlan, P. M. (1971), "Final Scientific Report on Grant AF EOAR-69-0049", With Computer Program QUINP, as an appendix.

5. O'Callaghan, M. J. (1972) "The Edge-Function Method for Linear Boundary Value Problems with Discontinuous Boundary Conditions", Ph.D. Thesis University College, Cork.

6. Quinlan, P. M. (1974), "The Edge-Function Method In Elastostatics" "Studies in Numerical Analysis", Festscript to Professor Lanczos, Academic Press.

7. Quinlan, P. M. (1974), "The Edge-Function Method" 16th British Colloquium Theoretical Mechanics, St. Andrews.

8. Desmukh, R. S. (1973), "The Edge-Function Method Applied to Moderately Thick Plates and Shallow Shells", Ph.D. Thesis, University of Massachusetts.

9. Quinlan, P. M. (1975), "The Edge-Function Method for Cracks and Stress Concentrations in Two Dimensional Problems"; To be published in Int. J. Num. Math. Eng.

10. Tai, I. H. and Nash, W. A. (1973), "Vibrations of Thin-Plates - A New Approach", AFOSR TR-74-0789. University of Massachusetts.

11. O'Callaghan, M. J., Nash, W. A. and Quinlan, P. M. (1975) "Vibration Analysis of Thin Shallow Spherical Shells Using The Edge-Function Method"; AFOSR TR 72-2340.

12. Quinlan, P. M. (1965), "The λ-Method for Rectangular Plates, Proc. Roy. Soc., Vol. 288A.

13. Quinlan, P. M. (1965), "The λ-Method for Aelotropic Rectangular and Skew Plates", Proc. Roy. Irish Acad.

14. Sokolnikoff, I. S. "Mathematical Theory of Elasticity", McGraw-Hill, 1956.

APPENDIX A - LAPEX

Program LAPEX consists of a simple main program LAPEX with four subroutines EDGEF, QPOLAR, POLW and QSOLVE(NE). The program is described briefly at the end of Section 1 and is attached together with illustrative example (37) as Appendices A(1) and A(2).

The corresponding data cards are

(1) CONTROL CARD

L, Ns, Ndiv, Mdiv, Nprog

where

L = Truncation level

Ns = No sides

Ndiv = No.divisions used in integral evaluation

Mdiv = No.of check points on sides

Nprog = Example no.

(2) COORDINATES VERTICES OF POLYGON

x(j), y(j)

(3) COEFFICIENTS FOR BOUNDARY CONDITIONS (37) on sides

c(j,k) (k = 1,4) - for each side

(4) PRODUCTION

U_1, V_1, U_2, V_2, Mdiv,

for function u at Mdiv equidistant points on line (U_1,V_1) to (U_2,V_2).

It is urged that the reader should become fully conversant with LAPEX, before proceeding to its fuller development in LAPGEN in Appendix B(2).

(5) Two Minor Subroutines are included: QPOLAR, to determine polar coordinates, and GAUSS to determine division points and weight factors for both Harmonic Fitting and Gaussian Integration. The program LAPGEN follows with numerous explanatory captions, the references being to the main paper.

An illustrative example is appended, as Appendix B(2), to attached program LAPGEN, Appendix B(1). This deals with the torsion of a quadilateral section with an elliptical cavity, Fig. 7, for four different truncation levels.

An effort has been made to make the output in Appendix B(2) self explanatory. The Boundary residuals are less than 1%, and the differences between computed quantities for levels LL = 2 and LL = 4 are seen to be considerably less.

Solutions of acceptable engineering accuracy are provided by truncation level LL = 1, involving only 53 equations with a time requirement on any computer of less than 150% of the time it would require to solve 53 linear equations using Gaussian elimination.

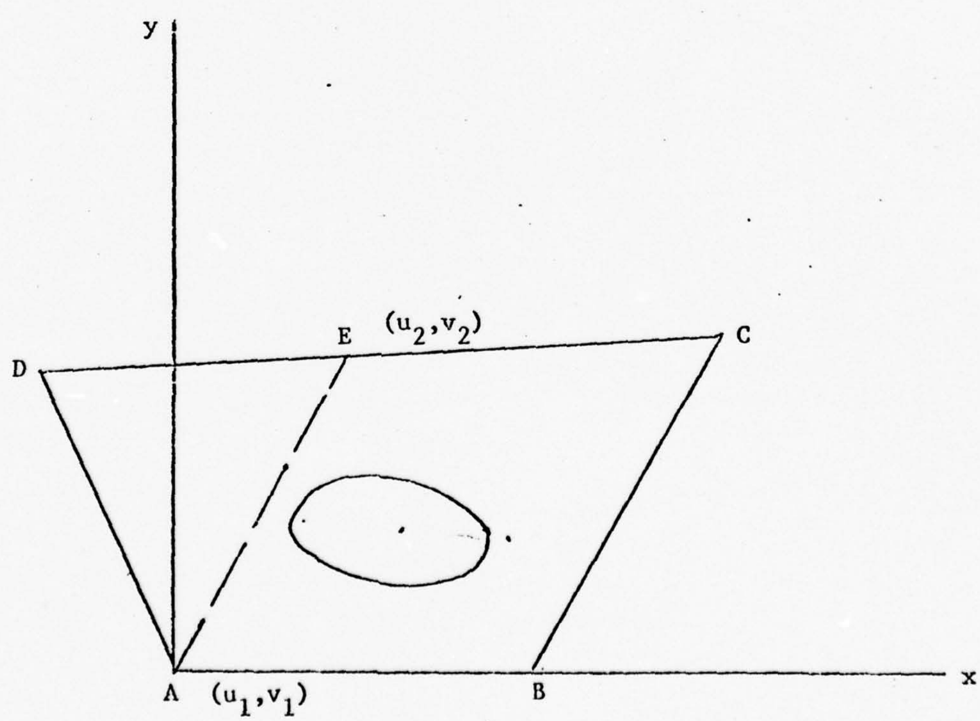


Figure 7.

APPENDIX B - LAPGEN

This program consists of main program LAPGEN and 10 subroutines consisting of

(1) COLMAT Sets up data for points at vertex j in the vertex equations taking care to eliminate any redundant equations. It then assigns the functions for solution "mix" (8), and sets up the necessary data for their evaluation at any point. Assigned points and functions are then printed out as a useful aid in checking.

(2) POINT FUNCTIONS for indicators MT = 1,2,3

(a) EDGEF Evaluates coeffs of unknowns E^1_{mj} and E^2_{mj} for Edge-Functions in eqs (55) and (56).

(b) POLW Evaluates Vercex Function V_{kj} in eq (50) and also Log-Vertex Functions V^*_{kj}, eq (60), where they replace V_{kj}.

(c) POLC Evaluates Harmonic Polars, eq (58), including the zero polar log r.

(d) PMAP Evaluates curved Edge-Functions based on section 3 for mapping z = c cosh w as given by eq (89) including the special case of a solid ellipse.

(e) POLP Evaluates r.h.s. of matrix. Particular integrals not yet included. Provision is made for boundary conditions of types NBDY = 1 and NBDY = 2.

(3) SOLCOR Solver routine for LS truncation levels, based on single pivoting for columns using the relevant equations.

(4) PRODN Arranges for production, other than computation of results for MT = 1,2,3 as indicated by Ncode = 1. This is divided into two categories: Ncode = 2, and Ncode = 3 corresponding to differentiation and integration respectively as the main operations involved. The several cases in each category are designated by the indicator NOFN. Provision is made for torsional rigidity, shear stress, resultant stress and shear lines for the torsional problem developed in Sokolnikoff [14]. Other cases can be added as they arise, and thus a comprehensive program for Laplace/Poisson problems can be built up.

APPENDIX C - DATA INPUT FOR LAPGEN

The data sets required are

(1) CONTROL CARD

L, NS, NB, NPROG, NBDY, LS, NMAT, NPRIN, MDIV, DELTA, FSET, GMOD

where

L	=	Maximum no. harmonics - if set as percentage of optimum, program will set corresponding L.
NS	=	No. boundary segments
NB	=	No. closed boundaries
NPROG	=	No. assigned to problem
NBDY	⇒	Type indicator for non-zero boundary conditions
	=	1 indicates polynomial $\sum_{k=1}^{6} C(j,k)t^{k-1}$
	=	2 indicates torsion conditions as set in POLP.
LS	=	No. Comparative solutions required
NMAT	⇒	Matrix print out given if set NMAT = 1
NPRIN	⇒	Gives print out of residuals on boundary if set NPRIN = 1
MDIV	⇒	No. of checking points required on boundary
DELTA	=	Increment for use in numerical differentiation
FSET	=	Degree of smoothing required
GMOD	=	Torsional rigidity; if dealing with torsion problem read in value for GMOD.

Note L, LS, MDIV, DELTA and FSET: if any of these are left blank, program will provide an appropriate value. Consequently only NS and NB must be specified.

(2) DATA FOR EACH SEGMENT

MB(J), NHS(J), NBY(J), NTYP(J), X(J), Y(J)

where

MB(J)	=	Function indicator MT as defined in Table 2
NHS(J)	=	Proportionality factor to regulate no. fitting points per side, approximately proportional to length. If left blank, will be set to one.
NBY(J)	⇒	Indicates if set to one that non-zero boundary conditions of type NBDY = 1 occur on side j, and corresponding coeffs. must be read in as data set (4) under.

NTYP(J) ⇒ Boundary curve type indicator for segment J where:
= 0 indicates straight line
= 1 indicates curved indentation
= 2 indicates curved mound
= 3 indicates closed curve

X(J),Y(J) ⇒ Coordinates of vertex J, where appropriate

(3) ADDITIONAL DATA CARD FOR EACH CURVED SEGMENT

AE(J), BE(J), XE(J), YE(J), ZE(J),
where for elliptic, or circular boundaries
Semi-axes are AE, BE; centre (XE,YE) and inclination of major axis ZE (as multiple of $\pi/2$).

(4) NON-ZERO BOUNDARY COEFFS. FOR NBDY = 1

C(J,K), (K = 1,6), for sides J with indicator NBY(J) = 1.

(5) PRODUCTION - one card per production set, unless when NCODE = 2 or NCODE = 3 a second card must follow to describe the function that is sought.

U_1, V_1, U_2, V_2, BXK, MDIV, MT, J, NCODE, NOFN, NAXIS,

where production is for

(i) SEGMENT (U_1,V_1) to (U_2,V_2) or if J is specified it is for segment J of boundary

(ii) BXK = angle associated with production (i.e. shear axis) as a multiple of $\pi/2$.

(iii) MDIV = No. equidistant points at which results are required.

(iv) NCODE ⇒ Indicates type of production required
= 1 : Functions MT = 1,2,3 as in Table 2
= 2 : Functions involving some differentiations
= 3 : Functions involving integration

(v) NOFN ⇒ Indicates different functions available for each of the production types NCODE = 2 and NCODE = 3.

APPENDICES A(1) + A(2)

```
C$JOB
C PROGRAM  LAPEX FOR SIMPLE  LAPLACE   PROBLEM
C
      COMMON X(11),Y(11),C(10,4),A(10),B(10),ANG(10),E(100),RMS(40),
     1G(100,101) ,SOL(100)
      COMMON NFU(101),NA(101),L,NS,NDIV,MDIV,NPROC
      DIMENSION HOLD(80,81)
      DIMENSION DUM(10432),IJK(207)
      EQUIVALENCE(DUM(1),X(1)),(IJK(1),NFU(1))
      DO 1234 K=1,10432
 1234 DUM(K)=0
      DO 1235 K=1,207
 1235 IJK(K)=0
C
C     SECTION 1.  READ AND PRINT DATA
C
      WRITE(6,799)
      WRITE(6,7899)
 7899 FORMAT(1H ,5X, ' PRINT OUT OF DATA CARDS FOR HEAT FLOW IN A QUADR
     1LATERAL ',/,'      TEMPERATURES GIVEN ON EDGES ')
      PI=3.14159265
      READ(5,8000) L,NS,NDIV,MDIV,NPROG
      WRITE(6,5000) L,NS,NDIV,MDIV,NPROG
      DO 1 J =1,NS
      READ(5,8001) X(J),Y(J)
    1 WRITE(6,5001) X(J),Y(J)
      DO 2 J=1,NS
      READ(5,8001)(C(J,K),K=1,4)
    2 WRITE(6,5001)(C(J,K),K=1,4)
      WRITE(6,799)
 8000 FORMAT(20I3)
 8001 FORMAT(7E11.4)
 5000 FORMAT(1H ,20I3)
 5001 FORMAT(1H ,7E11.4)
      NE =L*NS +NS
      NEE =NE+1
      NFP =NS
      X(NS+1) =X(1)
      Y(NS+1) =Y(1)
C   --------IF NDIV & MDIV LEFT BLANK SET TO USUAL NOS.
      IF(NDIV)4,3,4
    3 NDIV=L+2
    4 IF(MDIV)6,5,6
    5 MDIV=2*L+3
    6 CONTINUE
C
C     SECTION 2. SIDE LENGTHS A(J),SLOPES B(J) AND ANG(J)
C
      DO 12 J =1.NS
      XA=X(J+1)-X(J)
      YA=Y(J+1)-Y(J)
      CALL QPOLAR(XA,YA,RR,BA,PI)
      B(J)=BA
   12 A(J) = RR
      DO 1550 J =1,NS
      IF(J-1)15,13,14
   13 ANG(1) =PI-B(1) +B(NS)
      GO TO 15
   14 ANG(J) =PI-B(J)+B(J-1)
   15 IF(ANG(J)-2*PI)1550,1550,1540
```

```
 1540 ANG(J)=ANG(J)-2*PI
 1550 CONTINUE
C
C  SECTION 3.   ARRANGE FUNCTIONS
C
      NP = 0
      DO 21 M =1,L
      DO 21 J =1,NS
      NP =NP +1
      NA(NP)=J
      NFU(NP) =1
   21 E(NP) =M*PI/A(J)
      DO 22 J=1,NS
      NP =NS*L+J
      NA(NP)=J
      NFU(NP)=4
   22 E(NP)=PI/ANG(J)
      NE = NP
      LP =NP +1
      NFU(LP)=3
      NA(LP) = 10
C
C     SECTION 4. SET UP COLUMNS OF  MATRIX G(I,J)
C     HARMONICS ON CYCLE I=1 AND PTS ON I=2
      DO 50 NP =1,LP
      JE =NA(NP)
      NFN =NFU(NP)
      EM=E(NP)
      XJE=X(JE)
      YJE=Y(JE)
      BJE=B(JE)
      DO 50 I=1,2
      DO 50 J =1,NS
      KK =NDIV-1
      BA=B(J)
      IF(I-1)31,31,30
   30 KK =1
   31 DO 50 K =1,KK
      DK =0
      IF(I-1)33,32,33
   32 DK =K-1
      DK=DK/KK
      XK=X(J)+DK*(X(J+1)-X(J))
      YK=Y(J)+DK*(Y(J+1)-Y(J))
      GO TO 34
   33 XK=X(J)
      YK=Y(J)
   34 IF(NFN-3)35,36,37
   35 CALL QEDGEF(XK,YK,XJE,YJE,BJE,EM,VAL)
      GO TO 38
   36 VAL =C(J,1)+C(J,2)*DK+C(J,3)*DK**2+C(J,4)*DK**3
      GO TO 38
   37 CALL QPOLW(XK,YK,XJE,YJE,BJE,EM,VAL,PI)
   38 IF(I-1)40,40,39
   39 JX =NS*L+J
      G(JX,NP)=VAL
      GO TO 50
   40 DO 41 M =1,L
      JR=(M-1)*NS+J
   41 G(JR,NP)=G(JR,NP)+VAL*SIN(PI*M*DK)
```

```
   50 CONTINUE
C
C        SECTION 5.  SOLVE MATRIX G(I,J)
C   NEXT PROCEED TO CHECK RESIDUALS AT PTS ON SIDES AND
C   GET THEIR R.M.S, RMS(J),WITH NCODE=0 SET NCODE=1 FOR PRODUCTION
C    AT LINE 99
         WRITE(6,799)
   342 DO 348 JR=1,NE
         DO 348 JC=1,NEE
   348 HOLD(JR,JC)=G(JR,JC)
         CALL QSOLVE(NE)
   352 WRITE(6,5019)
  5019 FORMAT(1H ,10X, ' PRINT OUT OF RESIDUALS FOR EQUATIONS IF NE<50 '
         DO 350 JR=1,NE
         XX=0.00E 00
         DO 349 JC=1,NEE
  349    XX=XX+HOLD(JR,JC)*SOL(JC)
   350 WRITE(6,5007) JR,XX
         WRITE(6,799)
  353    NSJ=NS
         NCODE=0
         LPP=LP
         MMDIV=MDIV-1
  200    WRITE(6,799)
         DO 80 J=1,NSJ
         YY=0
         DO 70 K=1,MDIV
         XX=0
         DK=K-1
         DK=DK/MMDIV
         IF(NCODE)201,201,202
  202    XK=U1 +DK*(U2-U1)
         YK=V1+DK*(V2-V1)
         GO TO 203
  201    XK=X(J)+DK*(X(J+1)-X(J))
         YK=Y(J)+DK*(Y(J+1)-Y(J))
  203    DO 68 NP=1,LPP
         JE=NA(NP)
         NFN=NFU(NP)
         EM=E(NP)
         XJE=X(JE)
         YJE=Y(JE)
         BJE=B(JE)
         IF(NFN-3)65,66,67
   65    CALL QEDGEF(XK,YK,XJE,YJE,BJE,EM,VAL)
         GO TO 68
    66 VAL=C(J,1)+C(J,2)*DK+C(J,3)*DK**2+C(J,4)*DK**3
         GO TO 68
   67    CALL QPOLW(XK,YK,XJE,YJE,BJE,EM,VAL,PI)
C
C   CUMULATE XX FOR KTH POINT ON SIDE  J
C
   68    XX=XX+VAL*SOL(NP)
         G(J,K)=XX
   70    YY=YY+XX**2
   80    RMS(J)=YY
         IF(NCODE) 204,204,205
  205    WRITE(6,5009)MDIV
  5009 FORMAT(1H ,4X,' TEMPERATURES AT M= ',I3, ' PTS ON LINE U1,V1 ')
         GO TO 206
```

```
 204  WRITE(6,5003)
 5003 FORMAT(1H ,4X,' ROOT MEAN SQR RESIDUALS ON SIDE J ')
      WRITE(6,5004)(J,J=1,NS)
 5004 FORMAT(1H ,3X,8(10X,I3))
      WRITE(6,5005)(RMS(J),J=1,NS)
 5005 FORMAT(1H ,4X,8(2X,E11.4))
 799  FORMAT(1H ,6X, ' ***************************************** ')
      WRITE(6,5006)MDIV
 5006 FORMAT(1H ,6X,' PRINT OF RESID AT MDIV=',I3, ' PTS ON SIDE J ',/
     1  (NOTE: EVERY SECOND VALUE SHOULD BE ZERO IF MDIV LEFT BLANK) '
      WRITE(6,5004)(J,J=1,NS)
 206  DO 90 K=1,MDIV
  90  WRITE(6,5007)K,(G(J,K),J=1,NSJ)
 5007 FORMAT(1H ,I4,2X,7(2X,E11.4))
      READ(5,8008)U1,V1,U2,V2,MDIV
      WRITE(6,799)
      IF(MDIV)100,100,99
  99  WRITE(6,5011)
 5011 FORMAT(1H ,10X, ' PRODUCTION AT POINTS ON GIVEN LINE ',/)
      WRITE(6,5010)
 5010 FORMAT(1H ,10X, 'PRINT OUT OF DATA CARD  U1,V1,U2,V2,M ',/)
      WRITE(6,5008)U1,V1,U2,V2,MDIV
      NCODE=1
      LPP=LP-1
      MMDIV=MDIV-1
      NSJ=1
      GO TO 200
 8008 FORMAT(4E11.4,I3)
 5008 FORMAT(1H ,13X,4E11.4,I3)
 100  CALL EXIT
      STOP
      END

      SUBROUTINE QPOLAR(XA,YA,RR,BA,PI)
      RR=SQRT(XA**2+YA**2)
      IF(ABS(XA)-0.000001)7,8,8
   7  XA=XA+0.000001
   8  BA=ATAN(YA/XA)
      IF(XA)9,9,10
   9  BA=BA+PI
      GO TO 12
  10  IF(YA)11,12,12
  11  BA=BA+2*PI
  12  RETURN
      END

      SUBROUTINE QEDGEF(XK,YK,XJE,YJE,BJE,EM,VAL)
      XXK=(XK-XJE)*COS(BJE)+(YK-YJE)*SIN(BJE)
      YYK=-(XK-XJE)*SIN(BJE)+(YK-YJE)*COS(BJE)
      VAL=EXP(-EM*YYK)*SIN(EM*XXK)
      RETURN
      END

      SUBROUTINE QPOLW(XK,YK,XJE,YJE,BJE,EM,VAL,PI)
      XA=XK-XJE
      YA=YK-YJE
      CALL QPOLAR(XA,YA,RR,BA,PI)
      BA=BA-BJE
C     ----------ENSURE ANGLE BA>0
      IF(BA-0.1E-06)28,29,29
```

```
   28 BA=BA+2*PI
   29 VAL=RR*EM*SIN(EM*BA)
      RETURN
      END

      SUBROUTINE QSOLVE(NE)
      COMMON X(11),Y(11),C(10,4),A(10),B(10),ANG(10),E(100),RMS(40),
     1G(100,101),SOL(100)
      COMMON NFU(101),NA(101),L,NS,NDIV,MDIV,NPROG
      LP=NE+1
      DO 60 JR=1,NE
      XX=G(JR,JR)
      XX=1/XX
      DO 52 JC=1,LP
   52 G(JR,JC)=G(JR,JC)*XX
      DO 60 JRR=1,NE
      IF(JRR-JR)53,60,53
   53 XG=G(JRR,JR)
      DO 54 JCC=1,LP
   54 G(JRR,JCC)=G(JRR,JCC)-XG*G(JR,JCC)
   60 CONTINUE
      DO 61 JR=1,NE
   61 SOL(JR)=G(JR,LP)
      WRITE(6,5002)NPROG
 5002 FORMAT(1H ,6X, ' SOLUTION VECTOR NPROG =',I3)
      DO 62 K=1,NE
   62 WRITE(6,5001)SOL(K)
      WRITE(6,799)
 5001 FORMAT(1H ,7E11.4)
  799 FORMAT(1H ,6X, ' ************************************** ' )
      SOL(LP)=-1
      RETURN
      END
```

```
      C$ENTRY
      ****************************************
      PRINT OUT OF DATA CARDS FOR HEAT FLOW IN A QUADRILATERAL
   TEMPERATURES GIVEN ON EDGES
 10   4   0   0 12
 0.0000E 00 0.0000E 00
 0.1000E 01 0.0000E 00
 0.1200E 01 0.1000E 01
-0.2000E 00 0.9000E 00
 0.0000E 00 0.1000E 01-0.1000E 01 0.0000E 00
 0.0000E 00 0.0000E 00 0.0000E 00 0.0000E 00
 0.0000E 00 0.0000E 00 0.0000E 00 0.0000E 00
 0.0000E 00 0.0000E 00 0.0000E 00 0.0000E 00
      ****************************************
      ****************************************
      SOLUTION VECTOR NPROG = 12
 0.2662E 00
 0.2023E-02
-0.3286E-01
 0.1710E-01
 0.3065E-02
 0.4452E-02
-0.1160E-02
-0.6627E-02
 0.1156E-01
-0.1502E-02
```

```
-0.8570E-03
 0.2122E-02
 0.1344E-02
 0.6985E-03
-0.1084E-02
-0.1299E-02
 0.2991E-02
-0.1387E-02
-0.6895E-03
 0.5526E-03
 0.7245E-03
 0.1698E-03
-0.6242E-03
-0.5971E-03
 0.1198E-02
-0.8392E-03
-0.4160E-03
 0.5790E-05
 0.3750E-03
 0.4651E-04
-0.3333E-03
-0.4517E-03
 0.4704E-03
-0.4280E-03
-0.1954E-03
-0.1480E-03
 0.1171E-03
 0.9718E-05
-0.1055E-03
-0.1898E-03
 0.6103E-02
 0.5069E-02
-0.1729E-02
 0.3992E-02
          ************************************
                PRINT OUT OF RESIDUALS FOR EQUATIONS IF NE<50
     1      -0.1717E-04
     2       0.1531E-05
     3       0.2519E-05
     4       0.1665E-05
     5      -0.3651E-06
     6      -0.2272E-06
     7      -0.5902E-07
     8       0.2161E-06
     9      -0.8605E-06
    10       0.2049E-06
    11       0.6138E-06
    12      -0.3725E-07
    13      -0.6706E-07
    14      -0.1341E-06
    15       0.2581E-06
    16       0.1267E-06
    17       0.0000E 00
    18       0.1155E-06
    19       0.1607E-07
    20       0.1502E-06
    21      -0.1118E-06
    22       0.1602E-06
    23      -0.1075E-06
    24       0.1509E-06
```

25	-0.4680E-07
26	0.9313E-07
27	0.4155E-07
28	0.6799E-07
29	0.2095E-08
30	-0.3725E-08
31	-0.2467E-07
32	0.4680E-07
33	-0.5355E-08
34	0.2678E-07
35	0.2968E-07
36	0.2445E-07
37	0.1141E-07
38	-0.4889E-08
39	-0.1483E-07
40	0.4784E-08
41	0.4098E-07
42	-0.2570E-06
43	0.1562E-06
44	0.1215E-06

****

****

ROOT MEAN SQR RESIDUALS ON SIDE J

1	2	3	4
0.1407E-04	0.1977E-03	0.9401E-05	0.1761E-05

PRINT OF RESID AT MDIV= 23 PTS ON SIDE J

(NOTE: EVERY SECOND VALUE SHOULD BE ZERO IF MDIV LEFT BLANK)

	1	2	3	4
1	0.4098E-07	-0.2570E-06	0.1562E-06	0.1215E-06
2	-0.3601E-02	0.3486E-02	0.3017E-02	0.8766E-03
3	-0.1371E-05	0.1006E-06	0.2033E-06	0.1766E-06
4	0.7303E-03	-0.5569E-03	-0.4892E-03	0.1432E-03
5	-0.1609E-05	0.4470E-07	0.3243E-06	0.1880E-06
6	-0.3209E-03	0.2344E-03	0.1939E-03	0.7926E-03
7	-0.1848E-05	0.4470E-07	0.3160E-06	0.2414E-06
8	0.1851E-03	-0.1448E-03	-0.1045E-03	-0.2162E-03
9	-0.1788E-05	0.1416E-06	0.2359E-06	0.3180E-06
10	-0.1335E-03	0.1127E-03	0.6717E-04	0.1083E-03
11	-0.2205E-05	0.1676E-06	0.2327E-06	0.3616E-06
12	0.1041E-03	-0.1047E-03	-0.4717E-04	-0.6162E-04
13	-0.2205E-05	0.9686E-07	0.3078E-06	0.3465E-06
14	-0.9924E-04	0.1147E-03	0.3617E-04	0.3357E-04
15	-0.2265E-05	0.1192E-06	0.3774E-06	0.2384E-06
16	0.1009E-03	-0.6323E-02	-0.2742E-04	-0.6463E-05
17	-0.1907E-05	0.1974E-06	0.3782E-06	0.1974E-06
18	-0.1324E-03	-0.6755E-02	0.2074E-04	-0.3259E-04
19	-0.1967E-05	0.1914E-06	0.2898E-06	0.1267E-06
20	0.2153E-03	-0.8380E-02	-0.7443E-05	0.1266E-03
21	-0.1311E-05	0.1399E-06	0.1927E-06	0.4470E-07
22	-0.5616E-03	-0.5416E-02	-0.5271E-04	-0.5134E-03
23	-0.2570E-06	0.1562E-06	0.1404E-06	0.4098E-07

****

PRODUCTION AT POINTS ON GIVEN LINE

PRINT OUT OF DATA CARD U1,V1,U2,V2,M

0.0000E 00 0.0000E 00 0.1000E 01 0.5000E 00 10

****

TEMPERATURES AT M= 10 PTS ON LINE U1,V1

1	0.4098E-07

```
 2      0.8386E-01
 3      0.1264E 00
 4      0.1408E 00
 5      0.1353E 00
 6      0.1165E 00
 7      0.9024E-01
 8      0.6177E-01
 9      0.3552E-01
10      0.1452E-01
      ******************************************

CORE USAGE          OBJECT CODE=    11832 BYTES,ARRAY AREA=    68476 BYTES,TOTAL AREA

DIAGNOSTICS          NUMBER OF ERRORS=        0, NUMBER OF WARNINGS=        0, NUMBE

COMPILE TIME=     0.00 SEC,EXECUTION TIME=      0.00 SEC,  WATFIV - JUL 1973 V1L4
```

APPENDICES B(1)+B(2

```
C$JOB
C        *****************************************************
C     LAPGEN PROGRAM FOR GENERAL LAPLACE PROBLEMS         APPENDIX B(1)
C        *****************************************************
      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,1C
     CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(5
     C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TUR(4)
      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
      COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
      COMMON NDK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50
      COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
      DIMENSION HOLD(40,41)
      DIMENSION PADDY(11492),IJK(470)
      EQUIVALENCE (PADDY(1),X(1)),(IJK(1),NFU(1))
      DO 1234 I=1,11492
      PADDY(I)=0.0E+00
 1234 CONTINUE
      DO 1235 I = 1,470
 1235 IJK(I)=0
C
C                     SECTION 1.        READ IN AND PRINT OUT DATA.
C
      WRITE(6,799)
      WRITE(6,799)
C     --------------- CONTROL CARD
      READ(5,3000)L,NS,NB,NPROG,NBDY,LS,NMAT,NPRIN,MDIV,DELTA,FSET,GMC
      WRITE(6,7990)NPROG
      WRITE(6,799)
      READ(5,7991)(XCAR(I),I=1,40)
      WRITE(6,4991)(XCAR(I),I=1,40)
      WRITE(6,799)
      WRITE(6,5101)
      WRITE(6,5000)L,NS,NB,NPROG,NBDY,LS,NMAT,NPRIN,MDIV,DELTA,FSET,GM
      PI=3.14159265
      WRITE(6,799)
      WRITE(6,4990)NS
      WRITE(6,5002)
      NCARDS=3
      DO 1 J = 1,NS
C     ------------------- SEGMENT DATA
      READ(5,7999)MB(J),NHS(J),NBY(J),NTYP(J),X(J),Y(J)
      NCARDS=NCARDS+1
C     -----------     NCARD TALLIES THE DATA CARDS AS READ IN.
    1 WRITE(6,4999)J,MB(J),NHS(J),NBY(J),NTYP(J),X(J),Y(J)
      DO 878 J=1,NS
C     ----------- IF NHS(J) LEFT BLANK SET TO 1
      IF(NHS(J))878,877,878
  877 NHS(J) = 1
  878 CONTINUE
  879 WRITE(6,799)
      INDIC=0
      DO 1233 J=1,NS
      IF(NTYP(J))1233,1233,1230
C     ----------------- ADDITIONAL DATA FOR CURVED SIDES
 1230 READ(5,7998)AE(J),BE(J),XE(J),YE(J),ZE(J)
      NCARDS=NCARDS+1
      IF(INDIC)1232,1232,1231
 1232 WRITE(6,799)
      WRITE(6,4992)
```

```
      INDIC=1
      WRITE(6,5103)
 1231 WRITE(6,4998)AE(J),BE(J),XE(J),YE(J),ZE(J)
      ZE(J)=0.5*ZE(J)
 1233 CONTINUE
C     ------- IF DELTA & FSET & NB LEFT BLANK SET TO USUAL VALUES
      IF(DELTA)1237,1236,1237
 1236 DELTA = 0.1000E-03
 1237 IF(FSET)1239,1238,1239
 1238 FSET = 1.0000
 1239 IF(NB)1240,1240,1241
 1240 NB = 1
 1241 INDIC=0
      IF(NBDY-2)1245,1244,1244
 1244 DO 1246 J=1,NS
 1246 NBY(J)=1
      GO TO 1247
 1245 DO 2 J=1,NS
      IF(NBY(J))2,2,3
C     ------------------  READ COEFFS. C(J,K),(K=1,6) FOR NBDY=1
    3 READ(5,8001)(C(J,K),K=1,6)
      NCARDS=NCARDS+1
      IF(INDIC)1243,1242,1243
 1242 INDIC=1
      WRITE(6,799)
      WRITE(6,4993)
 1243 WRITE(6,5001)J,(C(J,K),K=1,6)
    2 CONTINUE
      WRITE(6,799)
C     --------------------  NSS = NO. SHARP CORNERS ON BOUNDARY
C                           NOK(J) = NO. INTEGRATION PTS.
C                    NROW(J) = WHERE HARMS. FOR SEGMENT J BEGIN IN EACH SET
C                    NVER(J) = VERTEX NO. FOR VERTEX J
 1247 NSS=NS
      IF(NTYP(NS)-3)5,4,4
    4 NSS=NS-1
      IF(NS-2)5,10,5
   10 NSS=0
    5 NROW(1)=1
      NH=0
      DO 9 J=1,NS
      NOK(J)=2*L*NHS(J)*FSET+3-NTYP(J)/3
      NVER(J)=0
      NROW(J+1)=NROW(J)+2*NHS(J)
      NH=NH+2*NHS(J)
      IF(NTYP(J)-3)6,9,9
    6 JJ=J-1
      IF(J-1)7,7,8
    7 JJ=NSS
    8 NVER(J)=2*(MB(J)-1)+MB(JJ)
    9 CONTINUE
C
C               SECTION 2.        SIDE LENGTHS,A(J), SLOPES,B(1,J) & B(2,J)
C                                 VERTEX ANGLES,ANG(J),FOR POLYGON.
C                                 ENSURE ANG(J) IN INTERVAL (0,2*PI)
      IF(NSS)16,16,11
   11 X(NSS+1)=X(1)
      Y(NSS+1)=Y(1)
      DO 12 J = 1,NS
      IF(NTYP(J))1510,1510,1520
```

```
 1520 CALL ELPS(J,0,0)
      GO TO 12
 1510 XA=X(J+1)-X(J)
      YA=Y(J+1)-Y(J)
      CALL QPOLAR(XA,YA,RR,BA,PI)
      B(1,J)=BA
      B(2,J)=BA
      A(J)=RR
   12 CONTINUE
      DO 1560 J=1,NSS
      IF(J-1)15,13,14
   13 ANG(1)=PI-B(1,1)+B(2,NSS)
      GO TO 15
   14 ANG(J)=PI-B(1,J)+B(2,J-1)
   15 IF(ANG(J)-2*PI)1550,1550,1540
 1540 ANG(J)=ANG(J)-2*PI
 1550 IF(ANG(J))1555,1560,1560
 1555 ANG(J)=ANG(J)+2*PI
 1560 CONTINUE
C     ---------- ADJUST L & LS TO APPRORIATE VALUE IF NECESSARY
      IF(L-25)18,17,17
   17 LMAX=(100-NFP)/NS
      L=(LMAX*L)/100
   18 IF(L-2)19,20,20
   19 L=2
   20 IF(L-LS)21,2500,2500
   21 LS=L
 2500 IF(LS)22,22,23
   22 LS=2
      GO TO 20
   23 IF(LS-4)2501,2501,24
   24 LS=4
 2501 CONTINUE
C     ------------ CALL SUBROUTINE COLMAT TO SET UP FIXED POINTS
C                  AND ASSIGN THE CORRESPONDING COLUMNS OF MATRIX
   16 CALL COLMAT
C
C                            SECTION 3.  SET UP COLUMNS NP OF MATRIX; POINTS
C                                        ON CYCLE I=2 AND HARMONICS ON I=1.
C       EVALUATE INTEGRALS BY TRAPEZOIDAL RULE WITH NOK(J) PTS. PER SID
      NCODE=0
      JCC=0
      NPP=(NH*L)/2
      DO 51 NP=1,LP
      NCOLS=1
      IF(NP-NPP)25,25,26
   25 NCOLS=2
C     -----------   COLUMN DATA AS SET UP IN COLMAT
   26 JE=NA(NP)
      NFN=NFU(NP)
      EM=E(NP)
      XJE=X(JE)
      YJE=Y(JE)
      BJE=B(1,JE)
      JRO=NH*L
      JRP=JRO+NZERO
      DO 501 I=1,2
      JJ=NFP
      NNEL=0
      IF(I-1)28,27,28
```

```
   27 JJ=NS
   28 DO 500 J=1,JJ
      IF(I-1)31,30,31
   30 KK=NOK(J)
      CALL GAUSS(KK,NCODE)
      MT=MB(J)
      JP=J
      GO TO 329
   31 KK=1
      M=LOC(J)
      MT=MBFIX(M)
      IF(MT-10)329,328,328
  328 KK=2
      MT=MT-10
  329 DO 50 K=1,KK
      IF(I-2)32,33,33
C     ---------- INTEGRATION POINTS AND WEIGHT FACTORS
   32 DK=D(K)
      WK=W(K)
      JNB=NTYP(J)
C
C     -----DATA FOR POINTS ON  ELLIPTIC  CURVE NO. NNEL
      IF(JNB)330,330,331
  331 IF(K-1)332,332,333
  332 IF(NP-1)333,334,333
  334 NNEL=NNEL+1
      NOEL(J)=NNEL
      CALL ELPS(J,NNEL,KK)
  333 NEL=NOEL(J)
      XK=XEL(NEL,K)
      YK=YEL(NEL,K)
      BXK=BEL(NEL,K)
      GO TO 34
C
C     ----DATA FOR POINTS ON STRAIGHT  LINE  SEGMENTS.
  330 XK=X(J)+DK*(X(J+1)-X(J))
      YK=Y(J)+DK*(Y(J+1)-Y(J))
      BXK=B(1,J)
      GO TO 34
C    -----DATA FOR FIXED POINTS AS ASSIGNED IN COLMAT.
   33 M=LOC(J)+K-1
      XK=XFIX(M)
      YK=YFIX(M)
      JP=JFIX(M)
      BXK=BFIX(M)
      DK=DFIX(M)
C     ----------- CALL  SUBROUTINES  FOR  POINT  VALUES
   34 GO TO (35,35,36,37,370,370),NFN
   35 CALL EDGEF(XK,YK,BXK,XJE,YJE,BJE,EM,PI,MT,VA,VB)
      GO TO 38
   36 CALL POLP(XK,YK,BXK,DK,MT,NCODE,JP,VA)
      NCOLS=1
      GO TO 38
   37 CA=COEF(1,NP)
      CB=COEF(2,NP)
      CALL POLW(XK,YK,BXK,XJE,YJE,BJE,EM,CA,CB,MT,VA,PI,DELTA)
      GO TO 38
  370 XJE=XE(JE)
      YJE=YE(JE)
      BJE=ZE(JE)
```

```
      IF(NFN-6)371,372,372
  372 CALL PMAP(XK,YK,BXK,XJE,YJE,BJE,EM,MT,JE,VA,VB)
      GO TO 38
  371 CALL POLC(XK,YK,BXK,XJE,YJE,BJE,EM,MT,VA,VB,PI)
   38 F(1)=VA
      F(2)=VB
C     --------  COMPUTE MATRIX ENTRIES AND INSERT IN G(JR,JX)
      IF(I-1)41,41,39
C     ---------             POINT OR 'TWO-POINT' EQS.
   39 JR=JRP+J
      DO 40 JC=1,NCOLS
      JX=JCC+JC
   40 G(JR,JX)=G(JR,JX)-F(JC)*((-1)**K)
      GO TO 50
   41 JR=JRO+J
C     ------        ZERO HARMONIC EQS.
      DO 42 JC=1,NCOLS
      JX=JCC+JC
   42 G(JR,JX)=G(JR,JX)+WK*F(JC)
      NN=0
      MM=NHS(J)
C     ------       COSINE AND SINE HARMONIC EQS.
      DO 44 MS=1,L
      DO 44 M=1,MM
      NN=NN+1
      DO 44 NEG=1,2
      BETA=(NEG-1)*PI*0.5E+00
      JR=(MS-1)*NH+NROW(J)+(2*(M-1))+NEG-1
      CC=COS(2*PI*DK*NN+BETA)
      DO 44 JC=1,NCOLS
      JX=JCC+JC
   44 G(JR,JX)=G(JR,JX)+WK*CC*F(JC)
   50 CONTINUE
  500 CONTINUE
  501 CONTINUE
   51 JCC=JCC+NCOLS
      NEE=NE+1
C     ---------- PRINT OUT OF COEFFS. MATRIX IF NMAT SET TO 1
      IF(NMAT)342,342,341
  341 WRITE(6,799)
      WRITE(6,5018)
      DO 347 JC=1,NEE
      WRITE(6,5016)JC
      WRITE(6,5017)(G(JR,JC),JR=1,NE)
  347 CONTINUE
      WRITE(6,799)
  342 IF(NE-40)343,343,351
  343 DO 348 JR=1,NE
      DO 348 JC=1,NEE
  348 HOLD(JR,JC)=G(JR,JC)
      WRITE(6,799)
C     ----------- CALL SUBROUTINE TO SOLVE MATRIX FOR LS SOLUTIONS
C                          AND CHECK RESULTS IF NE<41
  351 CALL SOLCOR
      IF(NE-40)352,352,353
  352 WRITE(6,799)
      WRITE(6,5019)LS
      WRITE(6,5004)(LL,LL=1,LS)
      DO 3500 JR=1,NE
      DO 350 LL=1,LS
```

```
      F(LL)=0.00E+00
      DO 349 JC=1,NEE
  349 F(LL)=F(LL)+HOLD(JR,JC)*SOL(LL,JC)
  350 CONTINUE
 3500 WRITE(6,5007)JR,(F(LL),LL=1,LS)
      WRITE(6,799)
C
C        -------- SECTION 4
C         NEXT PROCEED TO CHECK RESIDUALS AT POINTS K ON SIDES J,(J=1,NS)
C         AND GET THEIR R.M.S.,RMS(J),WITH NCODE=0 FOR LEVELS OF SOLUTION
C          LL=1,LS. PROGRAM SIMILAR TO COEFFS. MATRIX PROG. LABELS 25-40.
      WRITE(6,799)
  353 NSJ=NS
      NSA=1
      NSIDE=1
C     ----- INDICATORS INTRODUCED SO THAT CHECK PART OF PROGRAM CAN BE
C           USED AGAIN FOR PRODUCTION--SEE LABEL 2360.
      NCY=1
      NNHOLD=0
  200 WRITE(6,799)
      NNEL=0
      DO 80 NN=1,NCY
      DO 80 J=NSA,NSJ
      NOEL(J)=0
C     ----------- IF MDIV LEFT BLANK SET TO SUITABLE NO. TO CHECK AT
C                 'FIXED' POINTS AND MID-POINTS INBETWEEN
      IF(MDIV)354,354,355
  354 MDIV=2*NOK(J)-1
  355 DO 356 LL=1,LS
  356 RMS(LL,J)=0
      CALL GAUSS(MDIV,NCODE)
      JNB=NTYP(J)
      DO 221 K=1,MDIV
      G(1,K)=0
      G(2,K)=0
      G(3,K)=0
      G(4,K)=0
      DK=D(K)
      IF(NSIDE)202,202,201
  202 XK=U1+DK*(U2-U1)
      YK=V1+DK*(V2-V1)
      GO TO 210
  201 IF(JNB)209,209,199
  199 IF(K-1)203,203,206
  203 NNEL=NNEL+1
      NOEL(J)=NNEL
      CALL ELPS(J,NNEL,MDIV)
  206 NEL=NOEL(J)
      XK=XEL(NEL,K)
      YK=YEL(NEL,K)
      BXK=BEL(NEL,K)
      GO TO 208
  209 XK=X(J)+DK*(X(J+1)-X(J))
      YK=Y(J)+DK*(Y(J+1)-Y(J))
      BXK=B(1,J)
  208 IF(NCODE)204,204,210
  204 MT=MB(J)
  210 JCC=0
      IF(NCY-2)361,360,361
  360 BXK=BHOLD(NN)
```

```
  361 XFIX(K)=XK
      YFIX(K)=YK
      BFIX(K)=BXK
      DO 219 NP=1,LP
      JE=NA(NP)
      NFN=NFJ(NP)
      NCOLS=1
      IF(NP-NPP)211,211,212
  211 NCOLS=2
  212 EM=E(NP)
      XJE=X(JE)
      YJE=Y(JE)
      BJE=B(1,JE)
      GO TO (213,213,214,215,216,216),NFN
  213 CALL EDGEF(XK,YK,BXK,XJE,YJE,BJE,EM,PI,MT,VA,VB)
      GO TO 217
  214 CALL POLP(XK,YK,BXK,DK,MT,NCODE,J,VA)
      NCOLS=1
      GO TO 217
  215 CA=COEF(1,NP)
      CB=COEF(2,NP)
      CALL POLW(XK,YK,BXK,XJE,YJE,BJE,EM,CA,CB,MT,VA,PI,DELTA)
      GO TO 217
  216 XJE=XE(JE)
      YJE=YE(JE)
      BJE=ZE(JE)
      IF(NFN-6)2216,2217,2217
 2217 CALL PMAP(XK,YK,BXK,XJE,YJE,BJE,EM,MT,JE,VA,VB)
      GO TO 217
 2216 CALL POLC(XK,YK,BXK,XJE,YJE,BJE,EM,MT,VA,VB,PI)
  217 F(1)=VA
      F(2)=VB
C     ------ CUMULATE COEFFS. IN COL.JX FOR POINT K, WITH CORRESPONDING
C            SOLUTION SOL(LL,LX) AND FIND CORRESPONDING R.M.S. FOR EACH
C            SIDE J AND TRUNCATION LEVEL LL.
      DO 218 JC=1,NCOLS
      JX=JCC+JC
      DO 218 LL=1,LS
  218 G(LL,K)=G(LL,K)+F(JC)*SOL(LL,JX)
  219 JCC=JCC+NCOLS
      IF(NCODE)2219,2219,221
 2219 DO 220 LL=1,LS
  220 RMS(LL,J)=RMS(LL,J)+G(LL,K)**2
  221 CONTINUE
C
C                    SECTION 5.    PRINT OUT AND PRODUCTION ARRANGEMENTS
C     ------ IF NCODE>1 FURTHER PROCESSING AND PRINT OUT DONE IN
C                                                  SUBROUTINE PRODN.
      IF(NCODE-1)2222,235,2223
 2223 CALL PRODN(J,NSJ,NCODE,NOFN,NNHOLD,NHOLD)
      GO TO 80
 2222 DO 222 LL=1,LS
  222 RMS(LL,J)=SQRT(RMS(LL,J)/MDIV)
      GO TO 234
  235 WRITE(6,5009)MT,MDIV
      GO TO 236
  234 IF(NPRIN)80,80,2224
 2224 WRITE(6,799)
      WRITE(6,5006)MDIV,J
      NPRIN=NPRIN-1
```

```
  236 WRITE(5,5004)(LL,LL=1,LS)
      DO 79 K=1,MDIV
   79 WRITE(5,5007)K,(G(LL,K),LL=1,LS)
   80 CONTINUE
      IF(NCODE)2340,2340,2360
 2340 WRITE(6,799)
      WRITE(6,5003)
      WRITE(6,5004)(LL,LL=1,LS)
      DO 2350 J=1,NS
 2350 WRITE(6,5005)J,(RMS(LL,J),LL=1,LS)
C     -----DATA CARD FOR PRODUCTION NCODE = 1 INVOLVES COMPUTING
C          FUNCTIONS MT =1,2,3. EITHER COORDS.(U1,V1),(U2,V2) OF LINE, O
C            ITS SIDE NO.J (WHERE IT IS A BOUNDARY SEGMENT) ARE SPECIFIED
C     -----NCODE = 2 INVOLVES DIFFERENTIATION AND
C          NCODE = 3 INTEGRATIONS FOR OPERATION NOFN.
 2360 READ(5,8008)U1,V1,U2,V2,BXK,MDIV,MT,J,NCODE,NOFN,NAXIS
      NCARDS=NCARDS+1
      IF(MDIV+NCODE)100,100,99
   99 WRITE(6,799)
      IF(NCODE-1)2361,2361,2362
 2362 IF(NHOLD)2364,2364,2365
 2364 READ(5,7991)(XCAR(I),I=1,20)
      NCARDS=NCARDS+1
      WRITE(6,4991)(XCAR(I),I=1,20)
      GO TO 2365
 2361 WRITE(6,5011)
 2365 IF(NCODE-3)2369,2368,2369
 2368 IF(NOFN-1)2369,2370,2369
 2369 WRITE(6,5010)
      WRITE(6,5008)U1,V1,U2,V2,BXK,MDIV,MT,J,NCODE,NOFN,NAXIS
 2370 IF(NAXIS)2366,2366,2367
 2366 NAXIS=1
 2367 BXK=BXK*PI*0.5E+00
      NSIDE=J
      NSJ=1
      NCY=1
      NSA=1
      NHOLD=0
   92 IF(J)95,94,93
   94 IF(NCODE-2)95,490,96
   96 GO TO(97,98,98),NOFN
C     ----- NCODE = 3 ; NOFN = 1, TORSIONAL  RIGIDITY.
C                       NOFN > 1 AVAILABLE , NOT YET SPECIFIED.
   97 NSIDE=1
      NSJ=NS
      MT=1
      NHOLD=NS
      MDIV=24
      GO TO 95
   98 CONTINUE
      GO TO 95
  490 GO TO(492,491,493),NOFN
C     ------ NCODE = 2 ; NOFN = 1, SHEAR STRESS IN TORSION PROBLEM.
C                        NOFN = 2, RESULTANT STRESS,AND ITS DIRECTION;
C                                      INVOLVES TWO CYCLES NN =1,2.
C                        NOFN = 3, SHEAR LINES IN TORSION.
  491 BHOLD(1)=0
      BHOLD(2)=0.5*PI
      NCY=2
      MT=2
```

```
      NHOLD=2
      GO TO 95
  492 MT=2
      GO TO 95
  493 MT=3
      GO TO 95
   93 NSA=J
      NSJ=J
      GO TO 94
   95 GO TO 200
  100 WRITE(6,799)
C     ------------ FINAL OPERATION : PRINT OUT OF NCARD DATA CARDS FOR
C                                                          PROBLEM.
      WRITE(6,5120)NCARDS,NPROG
      DO 120 N=1,NCARDS
      READ(5,7991)(XCAR(I),I=1,20)
  120 WRITE(6,4991)(XCAR(I),I=1,20)
      WRITE(6,799)
C
C     ------------------- FORMAT STATEMENTS
C
 7990 FORMAT(1H ,10X,'APPENDIX B(2)   DATA PRINT OUT FOR NPROG =',I3,/)
 7991 FORMAT(20A4)
 4991 FORMAT(1H ,20A4)
 5101 FORMAT(1H ,'  L   NS   NB NPROG NBDY  LS NMAT NPRIN MDIV    DELTA
     1    FSET      GMOD')
 8000 FORMAT(9I3,3E11.4)
 5000 FORMAT(1H ,9(I3,2X),3E11.4)
 4990 FORMAT(1H ,10X,'DATA FOR NS =',I3,' SIDES',/)
 5002 FORMAT(1H ,'  J  MB(J)  NHS(J)    NBY(J) NTYP(J)     X(J)     Y(J)'
 4992 FORMAT(1H ,10X,'SUPPLEMENTARY DATA FOR ELLIP./CIRC. BDRIES',/)
 5103 FORMAT(1H ,'     AE(J)     BE(J)      XE(J)      YE(J)      ZE(J)')
 7998 FORMAT(5E11.4)
 4998 FORMAT(1H ,5E11.4)
 7999 FORMAT(4I3,3E11.4)
 4999 FORMAT(1H ,I4,4(I5,3X),3(E11.4,1X))
 4993 FORMAT(1H ,'J-SIDE COEFFS. C(J,K) FOR NON-ZERO BDRY. CONDITS. CAS
     1 NBRY=1')
 8001 FORMAT(7E11.4)
 5001 FORMAT(1H ,I4,3X,6(E11.4,1X))
 5018 FORMAT(1H ,6X, ' PRINT OF MATRIX BY COLS IF SET NMAT =1 ',/)
 5016 FORMAT(1H ,10X,'PRINT OUT OF COLUMN JC=',I3,/)
 5017 FORMAT(1X,8E11.4)
 5019 FORMAT(1H ,4X, 'PRINT OUT OF RESIDUAL FOR EQUATION NO. N IF NE<41
     C LS =',I3,'  TRUNC. LEVELS'/)
 5009 FORMAT(1H ,4X, 'FUNCTIONS MT =',I3,' AT N=MDIV = ',I3, 'PTS. ON L
     CNE (U1,V1),(U2,V2)  FOR TRUNC. LEVELS LL=1,LS ' )
 5006 FORMAT(1H ,3X,'PRINT OUT OF RESIDUALS AT MDIV=',I3,' POINTS ON SI
     CE J =',I3,' FOR TRUNC. LEVELS LL',/)
 5007 FORMAT(1H ,I4,2X,4(2X,E11.4))
 5003 FORMAT(1H ,6X, 'ROOT MEAN SQUARE OF BOUNDARY RESIDUALS ON SIDE N=
     C FOR TRUNC. LEVELS LL=1,LS ' )
 5004 FORMAT(1H ,' NO.=N',4(6X,'LL =',I3))
 5005 FORMAT(1H ,I4,4X,4(2X,E11.4))
 5011 FORMAT(1H ,10X,'PRODUCTION AT POINTS ON GIVEN LINE',/)
 5010 FORMAT(1H ,'DATA  U1          V1          U2          V2          BXK
     1        MDIV  MT   J   NCODE NOFN NAXIS')
 8008 FORMAT(5E11.4,5I3,I2)
 5008 FORMAT(1H ,4X,5E11.4,6I5)
  799 FORMAT(1H ,6X, ' ************************************** ' )
```

```
 5120 FORMAT(1H ,'  PRINT OUT OF NCARDS =',I3,' FORMATTED DATA CARDS FC
     1 CASE NPROG=',I3,/)
      STOP
      END
C
C
C
C
C
      SUBROUTINE COLMAT
      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
     CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50
     C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
      COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
      COMMON NOK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
      COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
C
C              SECTION 1.    TO SET UP DATA FOR POINTS AT VERTEX J IN TH
C                                       VERTEX EQUATIONS.
      NE=0
      NR=0
      NORF=0
      NP=NH*L/2
      NELIM=0
      JNORF=0
      IF(NSS)101,101,200
  200 DO 201 J=1,NSS
      NVEX=NVER(J)
      IF(NVEX)201,201,1
    1 DO 2 K=1,4
    2 NEQ(K)=1
      JOMIT=1
      TEST=ABS(COS(ANG(J)))
      IF(TEST-0.00001)3,3,4
    3 JOMIT=0
C     ---------EXAMINE THE VARIOUS TYPES , NVEX , OF CONDITIONS AT J.
C              MAXIMUM OF 4 VERTEX EQS.- IF REQUIRE TO OMIT ONE EQ. SET
C              CORRESPONDING INDICATOR NEQ(K) TO 0.
    4 GO TO (51,5,6,6),NVEX
   51 JJ=J-1
      IF(JJ)52,52,53
   52 JJ=NSS
   53 XX=C(JJ,1)+C(JJ,2)+C(JJ,3)+C(JJ,4)+C(JJ,5)+C(JJ,6)-C(J,1)
      IF(ABS(XX)-0.000001)54,54,55
   54 NEQ(3)=0
      GO TO 9
   55 NP=NP+1
C
C            DISCONTINUITY IN FUNCTION (NVEX=1) AT ANY VERTEX REQUIRES
C            USE OF CORRECTIVE THETA FUNCTION IN COL NP OF MIX AND SET UI
C            CORR. COL. DATA INDICATED BY SETTING E(NP)=2001. THESE ARE
C            TALLIED BY JNORF.
C
      NA(NP)=J
      NFU(NP)=4
      E(NP)=2001
      JNORF=JNORF+1
```

```
      GO TO 9
    5 NEQ(2)=JOMIT
      GO TO 9
    6 NEQ(4)=JOMIT
    9 DO 20 K=1,4
      IF(NEQ(K))20,20,10
   10 NR=NR+1
      NE=NE+1
      LOC(NE)=NR
      JJ=J
      GO TO (11,83,13,13),K
C     ------SET UP DATA FOR POINT,(K=1,K=3) AND 'TWO-POINT',(K=2,K=4)
C           VERTEX EQ. NE IN SUBSCRIPTED LOCATIONS NR.  NE TALLIES EQS.
   11 XFIX(NR)=X(J)
      YFIX(NR)=Y(J)
      BFIX(NR)=B(1,J)
      MBFIX(NR)=MB(J)
      JFIX(NR)=J
      DFIX(NR)=0
      GO TO 20
   13 JJ=J-1
      IF(JJ)14,14,15
   14 JJ=NSS
   15 IF(K-3)82,82,83
   82 XFIX(NR)=X(J)
      YFIX(NR)=Y(J)
      BFIX(NR)=B(2,JJ)
      JFIX(NR)=JJ
      MBFIX(NR)=MB(JJ)
      DFIX(NR)=1.0
      GO TO 20
   83 NR=NR-1
C     ------ 'TWO-POINT' EQS. ON STRAIGHT AND ELLIPTIC SEGMENTS.
      DO 100 M=1,2
      NR=NR+1
      DK=((-1)**M)*DELTA
      IF(K-2)85,85,84
   84 DK=1-DK
   85 DFIX(NR)=DK
      JFIX(NR)=JJ
      MBFIX(NR)=MB(JJ)+10
      IF(NTYP(JJ))86,86,87
   86 XFIX(NR)=X(JJ)+DK*(X(JJ+1)-X(JJ))
      YFIX(NR)=Y(JJ)+DK*(Y(JJ+1)-Y(JJ))
      BFIX(NR)=B(1,JJ)
C     -----------  SIMILAR  TO  SUBROUTINE  ELPS.
      GO TO 100
   87 Z1=PAR(1,JJ)+DK*(PAR(2,JJ)-PAR(1,JJ))
      ZZ=ZE(JJ)
      X1=COS(Z1)
      Y1=SIN(Z1)
      IF(Y1)89,88,89
   88 Y1=0.1D-10
   89 Z5=-BE(JJ)*X1/(AE(JJ)*Y1)
      ADD=PI
      IF(Z1-PI)91,91,90
   90 ADD=2*PI
   91 Z6=ATAN(Z5)+ADD
      ADD=0
      IF(NTYP(JJ)-2)93,92,93
```

```
   92 ADD=PI
   93 BFIX(NR)=Z6+ZZ+ADD
      X1=X1*AE(JJ)
      Y1=Y1*BE(JJ)
      XFIX(NR)=XE(JJ)+X1*COS(ZZ)-Y1*SIN(ZZ)
      YFIX(NR)=YE(JJ)+X1*SIN(ZZ)+Y1*COS(ZZ)
  100 CONTINUE
   20 CONTINUE
  201 CONTINUE
C              REMOVE INDETERMINACY IN NEUMANN PROBLEM BY PUTTING
C              FUNCTION TO ZERO AT LAST VERTEX, AND OMITTING THE LAST
C           NELIM EQS. AS LEFT BY COL. SELECTION PROCEDURE IN SOLCOR
  101 NELIM=1
      DO 103 J=1,NS
      IF(MB(J)-2)102,103,103
  102 NELIM=0
  103 CONTINUE
      IF(NELIM)105,105,104
  104 NR=NR+1
      NE=NE+1
      LOC(NE)=NR
      XFIX(NR)=X(2)
      YFIX(NR)=Y(2)+BE(1)
      MBFIX(NR)=1
      DFIX(NR)=0
      JFIX(NR)=10
C     -------- REMOVE INDETERMINACY IN HARMONIC CONJUGATE EPSI,MT=3, BY
C              SETTING EPSI=0 AT VERTEX 1, OR END OF MAJOR AXIS OF ELLIPS
  105 NR=NR+1
      NE=NE+1
      LOC(NE)=NR
      JFIX(NR)=10
      DFIX(NR)=0
      MBFIX(NR)=3
      XFIX(NR)=X(1)+AE(1)
      YFIX(NR)=Y(1)
  210 NFP=NE
      JRF=NR
      NZERO=NS
      NE=NE+NZERO
      NORF=NE-JNORF
C     ----------     NORF = TOTAL NO. OF VERTEX FUNCTIONS REQUIRED.
C                    NZERO = TOTAL NO. OF ZERO HARMONIC EQS.
      IF(NTYP(1))23,23,21
   21 NP=NP+1
C
C           SET UP COLS. FOR CONSTANT TERM FOR ZERO HARMONIC FOR EXTERIOR
C           ELLIPSE  & LOG FUNCTION FOR EACH ELLIPTIC INDENTATION
C                     INDICATED BY SETTING E(NP)=1000
      NA(NP)=1
      NFU(NP)=5
      E(NP)=0
      NORF=NORF-1
      IF(NS-2)24,23,23
   23 DO 240 J=2,NS
      IF(NTYP(J))240,240,230
  230 NP=NP+1
      NA(NP)=J
      NFU(NP)=5
      IF(AE(J)-BE(J)-0.100E-04)232,232,231
```

```
  231 NFU(NP)=6
  232 E(NP)=1000
      NORF=NORF-1
  240 CONTINUE
C
C               SECTION 2.       SET UP DATA FOR, AND ASSIGN, FUNCTIONS
C                         FOR COLUMNS OF MATRIX AS IN TABLE 1 -VERTEX
C                                                        FUNCTIONS.
C
   24 IF(NORF)37,37,25
   25 NNN=NORF/NSS
      NN=NORF-NNN*NSS
      DO 30 J=1,NSS
      NAD(J)=NNN
      IF(J-NN)29,29,30
   29 NAD(J)=NNN+1
   30 CONTINUE
      DO 40 J=1,NSS
C     ------   NAD(J) = NO. OF VERTEX FUNCTIONS REQUIRED FROM VERTEX J
      KK=NAD(J)
      DO 40 K=1,KK
      NP=NP+1
      NVEX=NVER(J)
      ALFA=ANG(J)
      EE=K*PI/ALFA
      EF=EE-PI/(2*ALFA)
      GO TO (31,32,33,34),NVEX
   31 CA=0
      CB=1
      CE=EE
      GO TO 35
   32 CA=0
      CB=1
      CE=EF
      GO TO 35
   33 CA=1
      CB=0
      CE=EF
      GO TO 35
   34 CA=1
      CB=0
      CE=EE
   35 COEF(1,NP)=CA
      COEF(2,NP)=CB
      NA(NP)=J
      E(NP)=CE
      NFU(NP)=4
C
C          CHECK FOR INTEGER VALUES OF LAMDA=CE, AND INDICATE BY
C           INCREASING STORAGE VALUE E(NP) BY 1000.
      NX=CE+0.0000001
      IF(CE-NX-0.0001)36,36,40
   36 E(NP)=CE+1000
   40 CONTINUE
C                 SET UP COLUMN LP FOR R.H.S. OF EQUATION.
      LP=NP+1
      NFU(LP)=3
      NA(LP)=10
   37 NP=0
C
```

```
C                 SECTION 3.      SET UP COLUMN SETS NP, EACH COMPRISING TW
C                                 COLS., MATCHING TWO EQS. FOR HARMONIC SETS M
C                                                         ON SIDE J .
C
      DO 50 MS=1,L
      DO 50 J=1,NS
      MM=NHS(J)
      DO 50 M=1,MM
      NP=NP+1
      ME=(MS-1)*MM+M
      NA(NP)=J
C     ------------- FULL   EDGE  FNS.(55)
      IF(NTYP(J))41,41,42
   41 NFU(NP)=1
      E(NP)=2*PI*ME/A(J)
      GO TO 44
C     -----------  POLAR  NFU = 5, OR CURVED EDGE FNS. NFU = 6
   42 NFU(NP)=5
      XX=ABS(AE(J)-BE(J))-0.1*AE(J)
      IF(XX)46,46,45
   45 NFU(NP)=6
   46 E(NP)=ME*2*PI/ABS(PAR(2,J)-PAR(1,J))
      IF(NTYP(J)-2)43,44,48
   48 IF(J-1)44,44,43
   43 E(NP)=-E(NP)
   44 NE=NE+2
   50 CONTINUE
C
C          PRINT OUT OF DETAILS FOR NFP FIXED POINTS AND LP COLUMN SETS.
C
      WRITE(6,902)NFP
      WRITE(6,799)
      WRITE(6,903)
      ZZ=NH*L+NZERO+0.5000
      DO 904 K=1,JRF
      ZZ=ZZ+1-0.4999*(MBFIX(K)/10)
      KK=ZZ
  904 WRITE(6,905)XFIX(K),YFIX(K),BFIX(K),DFIX(K),JFIX(K),MBFIX(K),KK
      WRITE(6,799)
      WRITE(6,906)LP
      WRITE(6,907)
      DO 908 K=1,LP
  908 WRITE(6,909)NFU(K),NA(K),E(K),K
      WRITE(6,799)
      WRITE(6,910)NE,NH,L
      WRITE(6,799)
  902 FORMAT(1H ,10X, ' DETAILS OF NFP=',I3,' FIXED  POINTS ASSIGNED IN
     1 COLMAT')
  799 FORMAT(11X, '****************************** ',/)
  903 FORMAT(7X, ' XFIX      YFIX      BFIX      DFIX           JFIX
     1MBFIX      EQ NO.')
  905 FORMAT(1X,4E11.4,3(7X,I3))
  906 FORMAT(7X,'DETAILS OF LP=',I3,' SETS OF FNS ASSIGNED IN COLMAT',/
  907 FORMAT(1X, '      NFU     NA    E(NP)        NP ' )
  909 FORMAT(1X,2(4X,I3),2X,E11.4,I5)
  910 FORMAT(1H ,4X,' NO. OF EQUATIONS IN MATRIX NE =',I3,/
     1     ,4X, 'NO. OF EQUATIONS IN HARMONIC SET NH =',I3,/
     2    ,4X, 'NO. OF HARMONIC SETS L= ',I3)
      RETURN
      END
```

```
C
C
C                    EDGEF.            EDGE FUNCTIONS,EQS.(55)+(56).
C
C           VA AND VB STORE COEFFS. OF UNKNOWNS FOR FUNCTION MT=1,2,OR 3.
C
      SUBROUTINE EDGEF(XK,YK,BXK,XJE,YJE,BJE,EM,PI,MT,VA,VB)
      XXK=(XK-XJE)*COS(BJE)+(YK-YJE)*SIN(BJE)
      YYK=-(XK-XJE)*SIN(BJE)+(YK-YJE)*COS(BJE)
      SSS=BXK-BJE
      EMM=EXP(-EM*YYK)
      IF(MT-2)1,2,4
    4 VA=EMM*SIN(EM*XXK)
      VB=EMM*COS(EM*XXK)
      GO TO 3
    1 VA=EMM*COS(EM*XXK)
      VB=-EMM*SIN(EM*XXK)
      GO TO 3
    2 VA=-EM*EMM*COS(EM*XXK+SSS)
      VB= EM*EMM*SIN(EM*XXK+SSS)
    3 RETURN
      END
C
C
C                    POLW.             VERTEX FUNCTIONS , TABLE 1 , VA STORES
C
C                              COEFF. (50) OF UNKNOWN .
C
      SUBROUTINE POLW(XK,YK,BXK,XJE,YJE,BJE,EM,CA,CB,MT,VA,PI,DELTA)
      XA=XK-XJE
      YA=YK-YJE
      CALL QPOLAR(XA,YA,RR,BA,PI)
C     -------- GIVES POLAR COORDS. (RR,BA) FOR POINT.
      BA=BA-BJE
      DDEL=10*DELTA
      IF(BA+DDEL)28,28,29
C     ---------- ENSURE POLAR  ANGLE BA>-DDEL
   28 BA=BA+2*PI
   29 SS=BXK-BJE
      IF(EM-2000)32,32,30
C     ---------- FUNCTION THETA  INDICATOR EM>2000
   30 VA=BA
      IF(MT-2)50,31,41
   41 VA=-ALOG(RR)
      GO TO 50
   31 VA=COS(SS-BA)/RR
      GO TO 50
   32 NPS=1+EM/1000
C     ----------- LOG-VERTEX FNS. EQS.(60)  INDIC. EM>1000 ; NPS=2
C                 GOT BY NUMERICAL DIFF.  INTERVAL  DELTA
C                 GIVES VERTEX FUNCTIONS WHEN NPS =1 .
      EM=EM-1000*(NPS-1)
      XX=0
      DO 10 K=1,NPS
      EN=EM
      IF(NPS-2)45,44,45
   44 EN=EM*(1+((-1)**K)*DELTA)
   45 EMM=EN-MT+1
```

50	0.00000E 00	0.00000E 00	0.65030E-04	0.66308E-04
51	0.00000E 00	0.00000E 00	0.00000E 00	0.16098E-04
52	0.00000E 00	0.00000E 00	0.00000E 00	-0.16523E-03
53	0.00000E 00	0.00000E 00	0.00000E 00	0.40161E-04

```
      IF(MT-3)33,36,36
   36 EMM=EN-MT+3
   33 RE=0
      IF(RR-0.00001)20,20,19
   19 RE=RR**EMM
   20 IF(MT-2)1,2,3
    3 XX=XX+((-1)**K)*(CA*SIN(EM*BA)-CB*COS(EM*BA))*RE
      GO TO 10
    1 XX=XX+((-1)**K)*(CA*COS(EM*BA)+CB*SIN(EM*BA))*RE
      GO TO 10
    2 XX=XX+((-1)**K)*(-CA*SIN(EMM*BA+SS)+CB*COS(EMM*BA+SS))*EN*RE
   10 CONTINUE
      VA=XX
   50 RETURN
      END
C
C
C                 PMAP     MAPPING Z=C COSH(W)    CURVED EDGE-FUNCTIONS SEC.
C        VA AND VB STORE COEFFS. OF UNKNOWNS IN EQ(89) FOR MT=1,2,OR 3.
C

      SUBROUTINE PMAP(XK,YK,BXK,XJE,YJE,BJE,EM,MT,JE,VA,VB)
      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
     CS(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50
     C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
      COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
      COMMON NOK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
      COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
      XX=XK-XJE
      YY=YK-YJE
      ZK=BJE
      VBB=0
      VAA=0
C     ----------  EQUATIONS (82) , (83) , (84) .
      XA=XX*COS(ZK)+YY*SIN(ZK)
      YA=-XX*SIN(ZK)+YY*COS(ZK)
      CC=AE(JE)**2-BE(JE)**2
      C1=(CC-XA**2-YA**2)
      T=(-C1+SQRT(C1**2+4*CC*(YA**2)))/(2*CC)
      IF(ABS(YA)-0.1E-06)480,480,482
  480 T=-1+XA*XA/CC
      IF(T)481,481,482
  481 T=0
  482 T1=SQRT(T)+SQRT(1+T)
      XI=ALOG(T1)
      XA=XA*(EXP(2*XI)-1)
      YA=YA*(EXP(2*XI)+1)
      CALL QPOLAR(XA,YA,RR,BA,PI)
      ETA=BA
C
C     ------ SET UP EQS(89)+(90) FOR SOLID ELLIPSE.
      VA=0
      VB=0
      NCY=1
      IF(JE-1)1,1,3
    1 IF(NTYP(1)-3)3,2,3
    2 NCY=2
    3 DO 510 N=1,NCY
```

```
      IF(EM-999)6,6,490
  490 IF(MT-2)491,502,492
  491 VAA=ALOG(XI)
      GO TO 504
  492 VAA=BA
      GO TO 504
    6 IF(N-2)5,4,4
    4 EM=-EM
    5 ETAM=BA*EM
      XIM=XI*EM
      FA=EXP(XIM)
      FB=COS(ETAM)
      FC=SIN(ETAM)
      IF(MT-2)501,502,503
  501 VAA=FA*FB
      VBB=-FA*FC
      GO TO 504
  502 X1=SIN(ETA)
      X2=COS(ETA)
      X3=EXP(XI)
      X4=1/X3
      X5=X3-X4
      X6=X3+X4
      DX=0.5*(X6*X6-4*X2*X2)
      IF(ABS(DX)-0.1E-07)505,505,506
  505 DX=0.1E-07
  506 CD=SQRT(CC)
      D1=X5*X2/(CD*DX)
      D2=-X6*X1/(CD*DX)
      IF(EM-999)494,494,493
  493 X2=COS(BXK)
      X1=SIN(BXK)
      X3=XI*XI+ETA*ETA
      X4=XI/X3
      X5=ETA/X3
      VAA=-X1*(X4*D1+X5*D2)+X2*(-X4*D2+X5*D1)
      GO TO 504
  494 FQJ=BXK-BJE
      FB=COS(ETAM-FQJ)
      FC=SIN(ETAM-FQJ)
      FA=FA*EM
      VAA=FA*(-D1*FC+D2*FB)
      VBB=FA*(-D1*FB-D2*FC)
      GO TO 504
  503 VAA=FA*FC
      VBB=FA*FB
  504 VA=VA+VAA
  510 VB=VB+VBB
      RETURN
      END
C
C
C            QPOLAR.      GIVES POLAR CORRDS.(RR,BA) WHERE BA CONTINUOUS
C                                  IN (0,2*PI).
C

      SUBROUTINE QPOLAR(XA,YA,RR,BA,PI)
      RR=SQRT(XA**2+YA**2)
      IF(ABS(XA)-0.00001)7,8,8
    7 XA=XA+0.00001
```

```
      8 BA=ATAN(YA/XA)
        IF(XA)9,9,10
      9 BA=BA+PI
        GO TO 12
     10 IF(YA)11,12,12
     11 BA=BA+2*PI
     12 RETURN
        END
C
C
C
C                  ELPS        GIVES  (1) PARAMETERS PAR(1,J)AND PAR(2,J) AND
C                           SLOPES B(1,J) AND B(2,J) AT END POINTS OF BDRY.
C                           SEGMENT FOR KK=0 ;
C                                     (2) COORDS.(XK,YK) AND SLOPE BXK AT KK
C                           POINTS ON SEGMENT,KK>0. STORED IN XEL,YEL,BEL.
C
        SUBROUTINE ELPS(J,NNEL,KK)
        COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
      CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50
      C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
        COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
        COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
        COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
        COMMON NOK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
        COMMON NFP,NZERO,NSS,NB,NDEL(10),LOC(50),NBDY,NFPOL
        IF(KK)300,300,330
C       ------------- PARAMETERS  OF  END  POINTS.
    300 IF(NTYP(J)-3)302,301,301
    301 PAR(1,J)=0
        PAR(2,J)=2*PI
        A(J)=2*PI
        GO TO 500
    302 IF(BE(J))303,303,304
    303 BE(J)=AE(J)
    304 DO 315 K=1,2
        JJ=J+K-1
        XX=X(JJ)-XE(J)
        YY=Y(JJ)-YE(J)
        ZK=ZE(J)
        XA=XX*COS(ZK)+YY*SIN(ZK)
        YA=-XX*SIN(ZK)+YY*COS(ZK)
        XXA=BE(J)*XA
        YYA=AE(J)*YA
        CALL QPOLAR(XXA,YYA,RR,BA,PI)
        XA=COS(BA)
        YA=SIN(BA)
        IF(YA)433,432,433
    432 YA=YA+0.000001
    433 BAA=ATAN(-BE(J)*XA/(AE(J)*YA))
        IF(YA)435,435,434
    434 BAA=BAA+PI
        GO TO 437
    435 IF(XA)436,436,437
    436 BAA=BAA+2*PI
    437 PAR(K,J)=BA
        B(K,J)=BAA+ZK
        IF(K-2)315,311,311
    311 PP=PAR(2,J)-PAR(1,J)
```

```
      IF(NTYP(J)-2)315,312,314
  312 IF(PP)315,315,313
  313 PAR(2,J)=PAR(2,J)-2*PI
      GO TO 315
  314 IF(PP)316,315,315
  316 PAR(2,J)=PAR(2,J)+2*PI
  315 CONTINUE
      A(J)=PAR(2,J)-PAR(1,J)
      GO TO 500
C     ------------ KK  POINTS  ON SEGMENT  J.
  330 DO 340 K=1,KK
      DK=D(K)
      XX=PAR(1,J)+DK*(PAR(2,J)-PAR(1,J))
      XA=COS(XX)
      YA=SIN(XX)
      IF(YA)333,332,333
  332 YA=YA+0.000001
  333 BA=ATAN(-BE(J)*XA/(AE(J)*YA))
      IF(YA)335,335,334
  334 BA=BA+PI
      GO TO 337
  335 BA=BA+2*PI
  337 CONTINUE
      ZK=ZE(J)
      XA=XA*AE(J)
      YA=YA*BE(J)
      XEL(NNEL,K)=XE(J)+XA*COS(ZK)-YA*SIN(ZK)
      YEL(NNEL,K)=YE(J)+XA*SIN(ZK)+YA*COS(ZK)
      BEL(NNEL,K)=BA+ZK
  340 CONTINUE
  500 RETURN
      END
C
C
C          SOLCOR.           SOLVER ROUTINE FOR LS TRUNCATION LEVELS.
C

      SUBROUTINE SOLCOR
      DIMENSION JRSTOP(100),HOLD(40,40)
      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
     CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50
     C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
      COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
      COMMON NOK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
      COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
      NDE=2
      LROW=NE-NH*L
      LLROW=LROW+1
      LSET=L*(NH/NDE)
      NEE=NE+1
C     ------------ MATRIX ARRANGED IN LSET SETS OF PAIRS OF HARMONIC
C                  EQS. FOR SUCCESSIVE SIDES IN EACH HARMONIC SET ; FOLLOWED
C                  BY LROW OTHER (ZERO HARMONICS AND VERTEX) EQS.. THE LAST
C                  ROW OF SET CONTAINING EQ. NO. MM IS JRSTOP(MM).
C
      MM=0
      DO 3 K=1,LSET
      DO 3 M=1,NDE
```

```
      MM=MM+1
    3 JRSTOP(MM)=NDE*K
      MMA=MM+1
      DO 4 MM=MMA,NE
    4 JRSTOP(MM)=NE
      KS=L-LS+1
      JR2=0
      NEQH=NH*L
      LLROW=LROW+1
C     ----------  SOLUTION FOR TRUNCATION AT KL HARMONICS
      DO 300 KL=KS,L
      JR1=JR2+1
      JR2=KL*NH
      JR3=JR2+LROW-1
      DO 240 JRA=JR1,JR3
C     ------------- ELIM. OF UNKNOWN JC TO TRIANGULATE MATRIX USING
C             PIVOT SELECTION FOR ROWS JC TO JRS--APPROPRIATE ROWS FOR
C             PIVOTING FOR COL JC .
      JC=JRA
      IF(JRA-JR2-1)201,190,200
  190 DO 191 JCC=1,LLROW
      DO 191 JRR=1,LROW
      JCX=NEQH+JCC
      JRX=NEQH+JRR
C     ------- HOLD AS REQUIRED WHEN SOLVING FOR NEXT KL TRUNC. LEVEL.
  191 HOLD(JRR,JCC)=G(JRX,JCX)
  200 JC=NEQH+JRA-JR2
  201 XX=0
      JRS=JRSTOP(JC)
      DO 210 JR=JC,JRS
      YY=ABS(G(JR,JC))
      IF(XX-YY)202,210,210
  202 XX=YY
      JRM=JR
      ZZ=G(JR,JC)
  210 CONTINUE
      DO 216 JCC=JC,NEE
      XX=G(JC,JCC)
      G(JC,JCC)=G(JRM,JCC)
  216 G(JRM,JCC)=XX
      JC1=JC+1
      ZZ=1/ZZ
      DO 220 JCC=JC,NEE
  220 G(JC,JCC)=G(JC,JCC)*ZZ
      DO 230 JRR=JC1,NE
      XG=G(JRR,JC)
      DO 230 JCC=JC,NEE
  230 G(JRR,JCC)=G(JRR,JCC)-XG*G(JC,JCC)
  240 CONTINUE
C     ------------ BACK SUBSTITUTE IN TRIANG. MATRIX FOR SOLN. FOR
C                  TRUNC. AT KL HARMONICS.
      LL=KL-KS+1
      SOL(LL,NEE)=-1
      ZZ=ABS(G(NE,NE))
      IF(ZZ-0.1000E-06)1240,1240,1241
 1240 NELIM=1
      GO TO 222
 1241 SOL(LL,NE)=G(NE,NEE)/G(NE,NE)
C     ---------- OMITS LAST NELIM EQUATIONS ;  NELIM SET IN COLMAT
C                MEANS OF CUTTING OUT REDUNDANT EQS.
```

```
      IF(NELIM)229,229,222
  222 DO 223 K=1,NELIM
      M=NE-K+1
      SOL(LL,NE)=0
  223 G(M,M+1)=0
  229 JR3=JR3+1
      DO 250 K=2,JR3
      M=NE-K+1
      M1=M+1
      IF(M-NEQH)242,242,243
  242 M=M-NEQH+JR2
  243 XX=0
      DO 245 MM=M1,NEE
      MA=MM
      IF(MM-NEQH)241,241,245
  241 MA=MM-NEQH+JR2
  245 XX=XX+SOL(LL,MA)*G(M,MA)
      SOL(LL,M)=-XX
      IF(K-LROW)250,246,250
  246 DO 247 JCC=1,LLROW
      DO 247 JRR=1,LROW
      JCX=NEQH+JCC
      JRX=NEQH+JRR
  247 G(JRX,JCX)=HOLD(JRR,JCC)
  250 CONTINUE
  300 CONTINUE
C     ------------- PRINT OUT OF SOLNS.
  350 WRITE(6,799)
      WRITE(6,110)
      WRITE(6,799)
      WRITE(6,111)LS
      WRITE(6,112)
      WRITE(6,115)(LL,LL=1,LS)
      DO 351 J = 1,NE
      WRITE(6,101)J,(SOL(K,J),K=1,LS)
  351 CONTINUE
  110 FORMAT(1H0,30X,'    SOLUTION VECTORS',/)
  111 FORMAT(1H ,'   SOLUTIONS OF COEFS. MATRIX FOR NO OF SETS LS =',I4
  112 FORMAT(1H ,'THE DIFF. LEVELS OF TRUNC. ARE DENOTED BY LL=1,LS')
  115 FORMAT(1H ,'COEF.',4(3X,' LL  =',I3,2X))
  101 FORMAT(1H ,I4,2X,4(D12.5,2X))
  799 FORMAT(1H ,25X,25H*************************//)
      RETURN
      END
C
C
C       POLC.                 HARMONIC  POLARS .
C

      SUBROUTINE POLC(XK,YK,BXK,XJE,YJE,BJE,EM,MT.VA,VB,PI)
      VA = 0
      VB = 0
      XA=XK-XJE
      YA=YK-YJE
      CALL QPOLAR(XA,YA,RR,BA,PI)
      SS = BXK-BJE
      BA=BA-BJE
      IF(EM-999)4,4,1
C     --------- LOG R   INDIC. EM=1000
    1 IF(MT-2)2,3,11
```

```
   11 VA=BA
      GO TO 9
    2 VA=ALOG(RR)
      GO TO 9
    3 VA=-SIN(SS-BA)/RR
      GO TO 9
    4 IF(EM)6,5,6
C     --------   CASE  EM = 0 , U = CONSTANT .
    5 VA=0
      IF(MT-1)9,15,9
   15 VA = 1
      GO TO 9
C     ------------ POLAR HARMONICS N & -N       EQS.(58).
    6 IF(MT-2)7,8,20
   20 XX=RR**EM
      VA=XX*SIN(EM*BA)
      VB=-XX*COS(EM*BA)
      GO TO 9
    7 XX=RR**EM
      VA=XX*COS(EM*BA)
      VB=XX*SIN(EM*BA)
      GO TO 9
    8 EMM = EM-1
      XX=(RR**EMM)*EM
      VA=-XX*SIN(EMM*BA+SS)
      VB=XX*COS(EMM*BA+SS)
    9 RETURN
      END
C
C
C        POLP      PARTICULAR SOLNS. (NOT YET INCLUDED), & R.H.S.
C

      SUBROUTINE POLP(XK,YK,BXK,DK,MT,NCODE,JP,VA)
      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
     CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50
     C, XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
      COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
      COMMON NOK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
      COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
      VA=0
      IF(NCODE)3,10,3
   10 J=JP
      IF(NBY(JP))3,3,1
    1 IF(NBDY-2)4,2,3
    4 VA=C(J,1)+C(J,2)*DK+C(J,3)*DK**2+C(J,4)*DK**3+C(J,5)*DK**4+C(J,6)*
     C*5
      GO TO 3
    2 IF(MT-2)3,5,3
    5 VA=-XK*COS(BXK)-YK*SIN(BXK)
C     --------- R.H.S. FOR TORSION PROBLEM
    3 RETURN
      END
C
C
C
C        PRODN     SUBROUTINE FOR SPECIAL OUTPUTS  NCODE=2 & NCODE=3,
C                    FOR CASES DESIGNATED BY NOFN = 1,2,3 .
```

```
C
      SUBROUTINE PRODN(J,NSJ,NCODE,NOFN,NNHOLD,NHOLD)
      COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10)
     CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50
     C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
      COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
      COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
      COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
      COMMON NOK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
      COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
      NNHOLD=NNHOLD+1
      LLA=1
      JUMP=0
      IF(NCODE-2)5,5,8
C     ---------------------- NCODE=2  MAY INVOLVE DIFFERENTIONS FOR
C                                   CASES NOFN = 1,2,3 GIVEN .
    5 DO 6 LL=1,LS
      DO 6 K=1,MDIV
      XK=XFIX(K)
      YK=YFIX(K)
      BXK=BFIX(K)
      GO TO(51,51,52),NOFN
   51 XX=1/(TOR(LL)*GMOD)
      XKG=COS(2*BXK)*XX
      XKF=(XK*COS(BXK)-YK*SIN(BXK))*XX
      NHOLD=NOFN
      IF(NOFN-2)54,53,54
   52 XKG=1
      XKF=-0.5*(XK*XK+YK*YK)
      NHOLD=1
      GO TO 54
   53 JLL=NNHOLD*4+LL
      GO TO 6
   54 JLL=LL
    6 G(JLL,K)=G(LL,K)*XKG+XKF
      IF(NHOLD-NNHOLD)20,7,20
    7 NNHOLD=0
      NHOLD=0
      GO TO(70,55,70),NOFN
C     ---------  PROCESSING FOR NOFN = 2 FOR RESULTANT STRESS AND ITS
C                                DIRECTION.
   55 DO 60 LL=1,LS
      DO 60 K=1,MDIV
      ZA=G(LL+4,K)
      ZB=G(LL+8,K)
      G(LL,K)=SQRT(ZA*ZA+ZB*ZB)
      ZC=ATAN(ZB/ZA)
      IF(ZC)56,60,60
   56 ZC=ZC+PI
   60 G(LL+12,K)=ZC*180/PI
      WRITE(5,790)
      JUMP=1
      WRITE(6,799)
      GO TO 70
   61 WRITE(6,791)
      JUMP=0
      LLA=13
      WRITE(6,799)
      GO TO 70
```

```
C        ------------------   RESULTS  PRINT  OUT
      70 WRITE(6,792)
         WRITE(6,793)(LL,LL=1,LS)
         DO 80 K=1,MDIV
         LLB=LLA+LS-1
      80 WRITE(6,794)XFIX(K),YFIX(K),(G(LX,K),LX=LLA,LLB)
C        ------------------ NCODE=3 INVOLVING INTEGRATIONS FOR FN. NOFN=1
C                  SPACE AVAILABLE FOR OTHER CASES DESIGNATED BY NOFN = 2,3
         IF(JUMP-1)20,61,20
       8 DO 15 K=1,MDIV
         XK=XFIX(K)
         YK=YFIX(K)
         BXK=BFIX(K)
         GO TO(11,12,13),NOFN
      11 IF(M3(J)-1)15,15,9
       9 C1=COS(BXK)
         S1=SIN(BXK)
         XKG=-XK*C1-YK*S1
         XKF=(S1*XK**3-C1*YK**3)/3.0
         WK=W(K)
         FW=1.0
         IF(J-1)18,18,16
      16 IF(NTYP(J)-3)18,17,18
      17 FW=-1.0
      18 WK=WK*0.5*A(J)*FW*NAXIS
         GO TO 13
      12 CONTINUE
      13 DO 14 LL=1,LS
      14 TOR(LL)=TOR(LL)+(G(LL,K)*XKG+XKF)*WK
      15 CONTINUE
         IF(NHOLD-NNHOLD)20,10,20
      10 NNHOLD=0
C        --------- PRINT OUT OF RESULTS FOR NCODE=3 AND NOFN=1.
         NHOLD=0
         WRITE(6,796)
         WRITE(6,797)
         WRITE(6,798)(TOR(LL),LL=1,LS)
     790 FORMAT(1H ,10X,'RESULTANT STRESSES AT SPECIFIED POINTS')
     791 FORMAT(1H ,4X,'DIRECTION OF RESULTANT STRESS WITH OX IN DEGREES')
     792 FORMAT(1H ,' POINT CO-ORDS. (XK,YK)  TRUNCATION LEVEL LL=1,LS')
     793 FORMAT(1H ,24X,4(5X,'LL =',I3))
     794 FORMAT(1H ,6(1X,E11.4))
     796 FORMAT(1H ,10X,'VALUE OF TORSIONAL RIGIDITY FOR TRUNC. LEVEL LL')
     797 FORMAT(1H ,'          LL=1          LL=2          LL=3          LL=4')
     798 FORMAT(1H ,4(2X,E11.4))
     799 FORMAT(1H ,6X,'****************************** ')
      20 RETURN
         END
C
C
C          SUBROUTINE GAUSS    FOR  PT. LOCATIONS D(K)  &  WT. FACTORS W(K
C

         SUBROUTINE GAUSS(N,NCODE)
         COMMON X(11),Y(11),C(10,6),A(10),B(2,10),ANG(10),E(100),RMS(4,10),
        CG(99,100),SOL(4,100),AE(11),BE(11),XE(11),YE(11),XFIX(50),YFIX(50)
        C,XEL(2,50),YEL(2,50),BEL(2,50),PAR(2,10),GMOD,DFIX(50),TOR(4)
         COMMON BFIX(50),ZE(11),F(4),COEF(2,100),PI,DELTA,XCAR(40),D(50)
         COMMON W(50),BHOLD(4),NFU(101),NA(101),L,NS,MDIV,NPROG,NE,LS
         COMMON NROW(11),NH,NAD(11),NVER(11),NEQ(4),LP,NAXIS,NELIM
```

```
      COMMON NOK(11),MB(11),NHS(11),NBY(11),NTYP(11),JFIX(50),MBFIX(50)
      COMMON NFP,NZERO,NSS,NB,NOEL(10),LOC(50),NBDY,NFPOL
      IF(NCODE-3)57,54,57
C
C     24 POINT SUBDIVISION           GAUSSIAN  QUADRATURE ;  NCODE = 3 .
C
   54 IF(N-24)57,55,57
   55 D(1) = .00240639
      D(2) = .01263572
      D(3) = .03086272
      D(4) = .05679223
      D(5) = .08999900
      D(6) = .12993790
      D(7) = .17595317
      D(8) = .22728926
      D(9) = .28310325
      D(10) = .34247866
      D(11) = .40444056
      D(12) = .46797156
      W(1) = .01234123
      W(2) = .02853139
      W(3) = .04427744
      W(4) = .05929858
      W(5) = .07334648
      W(6) = .08619016
      W(7) = .09761865
      W(8) = .10744427
      W(9) = .11550567
      W(10) = .12167047
      W(11) = .12583746
      W(12) = .12793820
      DO 56 K = 1,12
      K1 = 25-K
      D(K1) = 1.0-D(K)
   56 W(K1) = W(K)
      GO TO 62
C
C     EQUAL SUBDIVISIONS FOR HARMONIC FITTING ;  NCODE < 3.
   57 M = N-1
      Z1 = M
      Z1 = 1./Z1
      DO 58 K = 1,N
      Z2 = K-1
      D(K) = Z2*Z1
      W(K) = 2.0*Z1
      IF(K-1)59,59,60
   59 W(1) = Z1
      GO TO 58
   60 IF(K-N)58,61,61
   61 W(N) = Z1
   58 CONTINUE
   62 IF(N-50)901,901,902
  902 N = 50
   64 CONTINUE
  901 RETURN
      END

   C$ENTRY
****************************************
****************************************
```

APPENDIX B(2) DATA PRINT OUT FOR NPROG = 7

****
TORSION OF QUADRILATERAL PLATE WITH AN ELLIPTICAL HOLE LAP14880
ALL EDGES FULLY FIXED LAP14890
****

L	NS	NB	NPROG	NBDY	LS	NMAT	NPRIN	MDIV	DELTA	FSET	GMOD
6	5	2	7	2	4	0	1	0	0.0000E 00	0.0000E 00	0.1000E 05

****

DATA FOR NS = 5 SIDES

J	MB(J)	NHS(J)	NBY(J)	NTYP(J)	X(J)	Y(J)
1	2	1	0	0	0.0000E 00	0.0000E 00
2	2	1	0	0	0.1000E 01	0.0000E 00
3	2	1	0	0	0.1200E 01	0.1000E 01
4	2	1	0	0	-0.2000E 00	0.9000E 00
5	2	1	0	3	0.0000E 00	0.0000E 00

****
****

SUPPLEMENTARY DATA FOR ELLIP./CIRC. BDRIES

AE(J)	BE(J)	XE(J)	YE(J)	ZE(J)
0.2000E 00	0.1500E 00	0.6000E 00	0.5000E 00	0.0000E 00

DETAILS OF NFP= 18 FIXED POINTS ASSIGNED IN COLMAT

****

XFIX	YFIX	BFIX	DFIX	JFIX	MBFIX	EQ NO.
0.0000E 00	0.0000E 00	0.0000E 00	0.0000E 00	1	2	66
-0.1000E-03	0.0000E 00	0.0000E 00	-0.1000E-03	1	12	67
0.1000E-03	0.0000E 00	0.0000E 00	0.1000E-03	1	12	67
0.0000E 00	0.0000E 00	0.4931E 01	0.1000E 01	4	2	68
0.1979E-04	-0.8923E-04	0.4931E 01	0.1000E 01	4	12	69
-0.2003E-04	0.9000E-04	0.4931E 01	0.9999E 00	4	12	69
0.1000E 01	0.0000E 00	0.1373E 01	0.0000E 00	2	2	70
0.1000E 01	-0.1000E-03	0.1373E 01	-0.1000E-03	2	12	71
0.1000E 01	0.1000E-03	0.1373E 01	0.1000E-03	2	12	71
0.1000E 01	0.0000E 00	0.0000E 00	0.1000E 01	1	2	72
0.1000E 01	0.0000E 00	0.0000E 00	0.1000E 01	1	12	73
0.9999E 00	0.0000E 00	0.0000E 00	0.9999E 00	1	12	73
0.1200E 01	0.1000E 01	0.3213E 01	0.0000E 00	3	2	74
0.1200E 01	0.1000E 01	0.3213E 01	-0.1000E-03	3	12	75
0.1200E 01	0.1000E 01	0.3213E 01	0.1000E-03	3	12	75
0.1200E 01	0.1000E 01	0.1373E 01	0.1000E 01	2	2	76
0.1200E 01	0.1000E 01	0.1373E 01	0.1000E 01	2	12	77
0.1200E 01	0.9999E 00	0.1373E 01	0.9999E 00	2	12	77
-0.2000E 00	0.9000E 00	0.4931E 01	0.0000E 00	4	2	78
-0.2000E 00	0.9001E 00	0.4931E 01	-0.1000E-03	4	12	79
-0.2000E 00	0.8999E 00	0.4931E 01	0.1000E-03	4	12	79
-0.2000E 00	0.9000E 00	0.3213E 01	0.1000E 01	3	2	80
-0.2001E 00	0.9000E 00	0.3213E 01	0.1000E 01	3	12	81
-0.1999E 00	0.9000E 00	0.3213E 01	0.9999E 00	3	12	81
0.1000E 01	0.0000E 00	0.0000E 00	0.0000E 00	10	1	82
0.0000E 00	0.0000E 00	0.0000E 00	0.0000E 00	10	3	83

****

DETAILS OF LP= 54 SETS OF FNS ASSIGNED IN COLMAT

NFU	NA	E(NP)	NP
1	1	0.6283E 01	1
1	2	0.6161E 01	2

1	3	0.4477E	01	3
1	4	0.6815E	01	4
6	5	-0.1000E	01	5
1	1	0.1257E	02	6
1	2	0.1232E	02	7
1	3	0.8953E	01	8
1	4	0.1363E	02	9
6	5	-0.2000E	01	10
1	1	0.1885E	02	11
1	2	0.1848E	02	12
1	3	0.1343E	02	13
1	4	0.2045E	02	14
6	5	-0.3000E	01	15
1	1	0.2513E	02	16
1	2	0.2464E	02	17
1	3	0.1791E	02	18
1	4	0.2726E	02	19
6	5	-0.4000E	01	20
1	1	0.3142E	02	21
1	2	0.3081E	02	22
1	3	0.2238E	02	23
1	4	0.3408E	02	24
6	5	-0.5000E	01	25
1	1	0.3770E	02	26
1	2	0.3697E	02	27
1	3	0.2686E	02	28
1	4	0.4089E	02	29
6	5	-0.6000E	01	30
6	5	0.1000E	04	31
4	1	0.1756E	01	32
4	1	0.3511E	01	33
4	1	0.5267E	01	34
4	1	0.7022E	01	35
4	1	0.8778E	01	36
4	1	0.1053E	02	37
4	2	0.1777E	01	38
4	2	0.3553E	01	39
4	2	0.5330E	01	40
4	2	0.7107E	01	41
4	2	0.8884E	01	42
4	2	0.1066E	02	43
4	3	0.2413E	01	44
4	3	0.4825E	01	45
4	3	0.7238E	01	46
4	3	0.9651E	01	47
4	3	0.1206E	02	48
4	4	0.2207E	01	49
4	4	0.4414E	01	50
4	4	0.6621E	01	51
4	4	0.8828E	01	52
4	4	0.1104E	02	53
3	10	0.0000E	00	54

****

NO. OF EQUATIONS IN MATRIX NE = 83
NO. OF EQUATIONS IN HARMONIC SET NH = 10
NO. OF HARMONIC SETS L= 6

****

****

SOLUTION VECTORS

****

SOLUTIONS OF COEFS. MATRIX FOR NO OF SETS LS = 4
THE DIFF. LEVELS OF TRUNC. ARE DENOTED BY LL=1,LS

COEF.	LL = 1	LL = 2	LL = 3	LL = 4
1	0.98202E 00	0.98790E 00	0.10667E 01	0.10816E 01
2	-0.12209E 01	-0.10006E 01	-0.78331E 00	-0.68970E 00
3	-0.96259E 00	-0.34239E 00	-0.81472E 00	-0.76961E 00
4	-0.13042E 01	-0.14352E 01	-0.16078E 01	-0.16660E 01
5	-0.59695E 00	0.18468E 01	0.35011E 01	0.37497E 01
6	-0.14145E 02	-0.12207E 02	-0.10846E 02	-0.98086E 01
7	0.12652E 01	0.57255E 00	-0.14702E 00	-0.49969E 00
8	-0.94909E 00	-0.62094E 00	-0.29889E 00	-0.94453E-01
9	0.80594E-02	0.78857E-02	0.79028E-02	0.79200E-02
10	-0.40240E-01	-0.40287E-01	-0.40262E-01	-0.40210E-01
11	0.40181E-02	0.69710E-02	0.11042E-01	0.12609E-01
12	-0.18461E-01	-0.21662E-01	-0.24232E-01	-0.25972E-01
13	0.25959E-02	0.78425E-02	0.11652E-01	0.14338E-01
14	-0.21136E-01	-0.22965E-01	-0.25235E-01	-0.26269E-01
15	-0.11504E-01	-0.17149E-01	-0.30149E-01	-0.45522E-01
16	-0.15835E 00	-0.16781E 00	-0.18283E 00	-0.18570E 00
17	0.35026E-02	-0.43402E-02	-0.12983E-01	-0.16568E-01
18	-0.12326E-01	-0.87582E-02	-0.47824E-02	-0.21340E-02
19	-0.91725E-02	-0.91372E-02	-0.91465E-02	-0.91467E-02
20	-0.29329E-02	-0.27659E-02	-0.27954E-02	-0.28060E-02
21	0.19763E-03	0.47228E-03	0.84903E-03	0.10161E-02
22	-0.25906E-02	-0.34893E-02	-0.42350E-02	-0.46625E-02
23	0.90407E-03	0.16583E-02	0.22809E-02	0.27398E-02
24	-0.21414E-02	-0.23650E-02	-0.25742E-02	-0.26869E-02
25	-0.19492E-02	-0.33051E-02	-0.51392E-02	-0.72371E-02
26	-0.10626E-01	-0.13178E-01	-0.16012E-01	-0.17088E-01
27	-0.91864E-04	-0.77683E-03	-0.15673E-02	-0.18437E-02
28	-0.16595E-02	-0.14351E-02	-0.11194E-02	-0.92495E-03
29	-0.83256E-02	-0.83142E-02	-0.82783E-02	-0.82797E-02
30	-0.13836E-01	-0.13881E-01	-0.13883E-01	-0.13877E-01
31	0.00000E 00	0.65856E-04	0.13347E-03	0.17534E-03
32	0.00000E 00	-0.10163E-02	-0.12482E-02	-0.13767E-02
33	0.00000E 00	0.33821E-03	0.50581E-03	0.63882E-03
34	0.00000E 00	-0.52475E-03	-0.57492E-03	-0.60342E-03
35	0.00000E 00	-0.65517E-03	-0.10631E-02	-0.15626E-02
36	0.00000E 00	-0.21357E-02	-0.28411E-02	-0.31415E-02
37	0.00000E 00	-0.15067E-03	-0.29666E-03	-0.33667E-03
38	0.00000E 00	-0.43145E-03	-0.38806E-03	-0.36739E-03
39	0.00000E 00	-0.13468E-02	-0.13431E-02	-0.13385E-02
40	0.00000E 00	-0.12292E-01	-0.12297E-01	-0.12299E-01
41	0.00000E 00	0.00000E 00	0.28043E-04	0.45787E-04
42	0.00000E 00	0.00000E 00	-0.44818E-03	-0.49395E-03
43	0.00000E 00	0.00000E 00	0.11167E-03	0.16515E-03
44	0.00000E 00	0.00000E 00	-0.18122E-03	-0.19122E-03
45	0.00000E 00	0.00000E 00	-0.23341E-03	-0.41334E-03
46	0.00000E 00	0.00000E 00	-0.71644E-03	-0.81336E-03
47	0.00000E 00	0.00000E 00	-0.59593E-04	-0.64913E-04
48	0.00000E 00	0.00000E 00	-0.15110E-03	-0.15033E-03
49	0.00000E 00	0.00000E 00	-0.23869E-02	-0.23825E-02

50	0.00000E 00	0.00000E 00	0.65030E-04	0.66308E-04
51	0.00000E 00	0.00000E 00	0.00000E 00	0.16098E-04
52	0.00000E 00	0.00000E 00	0.00000E 00	-0.16523E-03
53	0.00000E 00	0.00000E 00	0.00000E 00	0.40161E-04
54	0.00000E 00	0.00000E 00	0.00000E 00	-0.60772E-04
55	0.00000E 00	0.00000E 00	0.00000E 00	-0.12581E-03
56	0.00000E 00	0.00000E 00	0.00000E 00	-0.22240E-03
57	0.00000E 00	0.00000E 00	0.00000E 00	-0.23417E-05
58	0.00000E 00	0.00000E 00	0.00000E 00	-0.54019E-04
59	0.00000E 00	0.00000E 00	0.00000E 00	-0.18667E-03
60	0.00000E 00	0.00000E 00	0.00000E 00	-0.17865E-04
61	-0.48392E-05	-0.33006E-05	-0.37253E-08	0.17136E-05
62	-0.64967E 00	-0.63527E 00	-0.64365E 00	-0.65110E 00
63	0.11853E 01	0.12693E 01	0.12626E 01	0.11622E 01
64	-0.11175E 01	-0.12530E 01	-0.12635E 01	-0.11532E 01
65	0.18150E 01	0.18471E 01	0.18632E 01	0.17811E 01
66	-0.46143E 00	-0.48994E 00	-0.51034E 00	-0.49356E 00
67	0.12514E 00	0.13851E 00	0.15257E 00	0.15437E 00
68	-0.53071E 00	-0.57256E 00	-0.59676E 00	-0.61077E 00
69	0.74352E 01	0.78382E 01	0.83432E 01	0.81987E 01
70	-0.46564E 01	-0.31518E 01	-0.20572E 01	-0.16009E 01
71	-0.13993E 01	-0.62454E 00	0.10731E-02	0.26625E 00
72	-0.91244E 00	-0.67918E 00	-0.50376E 00	-0.40875E 00
73	-0.10359E 00	-0.60605E-01	-0.22254E-01	-0.77015E-03
74	0.80109E 01	0.72289E 01	0.68057E 01	0.64581E 01
75	-0.34598E 00	-0.20922E 00	-0.10339E 00	-0.84905E-01
76	0.12713E 01	0.11016E 01	0.97750E 00	0.89642E 00
77	-0.30128E-01	-0.29446E-01	-0.30445E-01	-0.32503E-01
78	0.26388E-01	0.19068E-01	0.12103E-01	0.85009E-02
79	-0.57160E 01	-0.50588E 01	-0.46934E 01	-0.44428E 01
80	-0.27428E 01	-0.25124E 01	-0.24300E 01	-0.23495E 01
81	-0.74386E 00	-0.64149E 00	-0.58405E 00	-0.54462E 00
82	-0.19363E 00	-0.18339E 00	-0.18572E 00	-0.18225E 00
83	0.00000E 00	0.00000E 00	0.00000E 00	0.00000E 00

****

****

PRINT OUT OF RESIDUALS AT MDIV= 29 POINTS ON SIDE J = 1 FOR TRUNC. LEVELS LL

NO.=N	LL = 1	LL = 2	LL = 3	LL = 4
1	0.5311E-04	-0.1498E-03	-0.1801E-03	-0.3495E-03
2	0.3940E-01	0.3444E-01	0.2900E-01	0.2486E-01
3	0.2619E-01	0.1105E-01	0.2396E-02	-0.9006E-04
4	-0.1436E-02	-0.1296E-01	-0.1234E-01	-0.8153E-02
5	-0.1903E-01	-0.1479E-01	-0.4116E-02	0.3517E-03
6	-0.1929E-01	-0.1478E-02	0.6689E-02	0.4152E-02
7	-0.6915E-02	0.1010E-01	0.5720E-02	0.4178E-04
8	0.7640E-02	0.9735E-02	-0.2372E-02	-0.2242E-02
9	0.1515E-01	0.2779E-03	-0.5403E-02	0.3151E-03
10	0.1228E-01	-0.7916E-02	-0.4861E-03	0.1844E-02
11	0.2326E-02	-0.7284E-02	0.4691E-02	-0.3583E-04
12	-0.8443E-02	0.2494E-03	0.2923E-02	-0.1450E-02
13	-0.1294E-01	0.7238E-02	-0.2528E-02	0.2944E-03
14	-0.9037E-02	0.6882E-02	-0.4076E-02	0.1494E-02
15	0.6198E-03	-0.1469E-03	0.3118E-03	-0.8970E-04
16	0.9760E-02	-0.7015E-02	0.4268E-02	-0.1429E-02
17	0.1309E-01	-0.6632E-02	0.2415E-02	0.3245E-03
18	0.7390E-02	0.5659E-03	-0.3007E-02	0.1733E-02
19	-0.2834E-02	0.7825E-02	-0.4193E-02	-0.8345E-04
20	-0.1246E-01	0.7623E-02	0.1032E-02	-0.1896E-02
21	-0.1426E-01	-0.6124E-03	0.5673E-02	0.3985E-03

22	-0.6174E-02	-0.9458E-02	0.2240E-02	0.2795E-02
23	0.8077E-02	-0.9071E-02	-0.5546E-02	-0.1926E-03
24	0.1941E-01	0.2506E-02	-0.5902E-02	-0.4252E-02
25	0.1798E-01	0.1465E-01	0.4647E-02	0.2989E-03
26	0.4478E-03	0.1196E-01	0.1191E-01	0.8770E-02
27	-0.2529E-01	-0.1069E-01	-0.2365E-02	0.8261E-04
28	-0.3690E-01	-0.3176E-01	-0.2668E-01	-0.2419E-01
29	0.4882E-03	0.7324E-03	0.9612E-03	0.1038E-02

****

ROOT MEAN SQUARE OF BOUNDARY RESIDUALS ON SIDE N=J FOR TRUNC. LEVELS LL=1,

NO.=N	LL = 1	LL = 2	LL = 3	LL = 4
1	0.1585E-01	0.1172E-01	0.8747E-02	0.6980E-02
2	0.9986E-02	0.5661E-02	0.3932E-02	0.3123E-02
3	0.1942E-01	0.9760E-02	0.6485E-02	0.4515E-02
4	0.1045E-01	0.5795E-02	0.3882E-02	0.3332E-02
5	0.4041E-02	0.3850E-03	0.4144E-04	0.4069E-04

****

PRODUCTION AT POINTS ON GIVEN LINE

DATA U1	V1	U2	V2	BXK	MDIV	MT	J	NCO
0.0000E 00	0.0000E 00	0.5000E 00	0.9500E 00	0.0000E 00	11	1	0	1

****

FUNCTIONS MT = 1 AT N=MDIV = 11PTS. ON LINE (U1,V1),(U2,V2) FOR TRUNC. LE

NO.=N	LL = 1	LL = 2	LL = 3	LL = 4
1	-0.5676E 00	-0.5677E 00	-0.5674E 00	-0.5671E 00
2	-0.5754E 00	-0.5755E 00	-0.5753E 00	-0.5752E 00
3	-0.5895E 00	-0.5896E 00	-0.5894E 00	-0.5894E 00
4	-0.6056E 00	-0.6058E 00	-0.6056E 00	-0.6056E 00
5	-0.6227E 00	-0.6229E 00	-0.6227E 00	-0.6227E 00
6	-0.6409E 00	-0.6411E 00	-0.6410E 00	-0.6410E 00
7	-0.6612E 00	-0.6614E 00	-0.6613E 00	-0.6613E 00
8	-0.6829E 00	-0.6830E 00	-0.6829E 00	-0.6829E 00
9	-0.7060E 00	-0.7061E 00	-0.7060E 00	-0.7060E 00
10	-0.7337E 00	-0.7336E 00	-0.7336E 00	-0.7336E 00
11	-0.7686E 00	-0.7691E 00	-0.7688E 00	-0.7690E 00

****

PRODUCTION AT POINTS ON GIVEN LINE

DATA U1	V1	U2	V2	BXK	MDIV	MT	J	NCO
0.0000E 00	0.0000E 00	0.5000E 00	0.9500E 00	0.0000E 00	11	2	0	1

****

FUNCTIONS MT = 2 AT N=MDIV = 11PTS. ON LINE (U1,V1),(U2,V2) FOR TRUNC. LE

NO.=N	LL = 1	LL = 2	LL = 3	LL = 4
1	0.5311E-04	-0.1498E-03	-0.1801E-03	-0.3495E-03
2	-0.2970E 00	-0.2969E 00	-0.2973E 00	-0.2977E 00
3	-0.3895E 00	-0.3898E 00	-0.3901E 00	-0.3904E 00
4	-0.4306E 00	-0.4309E 00	-0.4311E 00	-0.4313E 00
5	-0.4476E 00	-0.4479E 00	-0.4480E 00	-0.4482E 00
6	-0.4555E 00	-0.4556E 00	-0.4555E 00	-0.4558E 00
7	-0.4613E 00	-0.4612E 00	-0.4613E 00	-0.4614E 00
8	-0.4639E 00	-0.4641E 00	-0.4641E 00	-0.4642E 00
9	-0.4743E 00	-0.4735E 00	-0.4737E 00	-0.4738E 00
10	-0.5050E 00	-0.5012E 00	-0.5025E 00	-0.5021E 00
11	-0.5465E 00	-0.5558E 00	-0.5518E 00	-0.5542E 00

****

SHEAR LINES CHI=CONST. CHI=EPSI-0.5*(XK*XK YK*YK) LAP14990

DATA U1	V1	U2	V2	BXK	MDIV	MT	J	NCO
0.0000E 00	0.0000E 00	0.5000E 00	0.9500E 00	0.0000E 00	11	3	0	2

****

POINT CO-ORDS. (XK,YK) TRUNCATION LEVEL LL=1,LS

		LL = 1	LL = 2	LL = 3	LL = 4
0.0000E 00	0.0000E 00	0.5722E-04	0.6104E-04	0.7534E-04	0.7153E-04
0.5000E-01	0.9500E-01	0.2181E-01	0.2227E-01	0.2244E-01	0.2247E-01
0.1000E 00	0.1900E 00	0.5765E-01	0.5807E-01	0.5821E-01	0.5823E-01
0.1500E 00	0.2850E 00	0.9274E-01	0.9318E-01	0.9331E-01	0.9333E-01
0.2000E 00	0.3800E 00	0.1211E 00	0.1216E 00	0.1217E 00	0.1217E 00
0.2500E 00	0.4750E 00	0.1387E 00	0.1393E 00	0.1394E 00	0.1394E 00
0.3000E 00	0.5700E 00	0.1419E 00	0.1424E 00	0.1426E 00	0.1426E 00
0.3500E 00	0.6650E 00	0.1276E 00	0.1283E 00	0.1285E 00	0.1285E 00
0.4000E 00	0.7600E 00	0.9699E-01	0.9777E-01	0.9789E-01	0.9792E-01
0.4500E 00	0.8550E 00	0.5123E-01	0.5180E-01	0.5197E-01	0.5200E-01
0.5000E 00	0.9500E 00	-0.9176E-02	-0.9931E-02	-0.9242E-02	-0.9396E-02

****

TORSIONAL RIGIDITY FOR TRUNC. LEVELS LL=1,LS LAP15010

****

VALUE OF TORSIONAL RIGIDITY FOR TRUNC. LEVEL LL

LL=1	LL=2	LL=3	LL=4
0.1654E 00	0.1653E 00	0.1653E 00	0.1652E 00

****

RESULTANT STRESS & ITS DIRECTION LAP15030

DATA	U1	V1	U2	V2	BXK	MDIV	MT	J	NCO
	0.0000E 00	0.0000E 00	0.5000E 00	0.9500E 00	0.0000E 00	11	0	0	2

****

RESULTANT STRESSES AT SPECIFIED POINTS

****

POINT CO-ORDS. (XK,YK) TRUNCATION LEVEL LL=1,LS

		LL = 1	LL = 2	LL = 3	LL = 4
0.0000E 00	0.0000E 00	0.5626E-07	0.1989E-06	0.2118E-06	0.3186E-06
0.5000E-01	0.9500E-01	0.1981E-03	0.1985E-03	0.1984E-03	0.1985E-03
0.1000E 00	0.1900E 00	0.2241E-03	0.2243E-03	0.2243E-03	0.2245E-03
0.1500E 00	0.2850E 00	0.2051E-03	0.2056E-03	0.2056E-03	0.2057E-03
0.2000E 00	0.3800E 00	0.1654E-03	0.1658E-03	0.1659E-03	0.1660E-03
0.2500E 00	0.4750E 00	0.1246E-03	0.1248E-03	0.1249E-03	0.1250E-03
0.3000E 00	0.5700E 00	0.1160E-03	0.1160E-03	0.1161E-03	0.1162E-03
0.3500E 00	0.6650E 00	0.1628E-03	0.1628E-03	0.1629E-03	0.1629E-03
0.4000E 00	0.7600E 00	0.2457E-03	0.2454E-03	0.2456E-03	0.2455E-03
0.4500E 00	0.8550E 00	0.3417E-03	0.3440E-03	0.3436E-03	0.3438E-03
0.5000E 00	0.9500E 00	0.4369E-03	0.4545E-03	0.4471E-03	0.4505E-03

DIRECTION OF RESULTANT STRESS WITH OX IN DEGREES

****

POINT CO-ORDS. (XK,YK) TRUNCATION LEVEL LL=1,LS

		LL = 1	LL = 2	LL = 3	LL = 4
0.0000E 00	0.0000E 00	0.1248E 03	0.1171E 03	0.1210E 03	0.1316E 03
0.5000E-01	0.9500E-01	0.1389E 03	0.1388E 03	0.1389E 03	0.1391E 03
0.1000E 00	0.1900E 00	0.1414E 03	0.1414E 03	0.1415E 03	0.1416E 03
0.1500E 00	0.2850E 00	0.1458E 03	0.1458E 03	0.1458E 03	0.1459E 03
0.2000E 00	0.3800E 00	0.1548E 03	0.1548E 03	0.1548E 03	0.1548E 03
0.2500E 00	0.4750E 00	0.1753E 03	0.1752E 03	0.1752E 03	0.1753E 03
0.3000E 00	0.5700E 00	0.3279E 02	0.3276E 02	0.3275E 02	0.3274E 02
0.3500E 00	0.6650E 00	0.6497E 02	0.6492E 02	0.6490E 02	0.6489E 02
0.4000E 00	0.7600E 00	0.7946E 02	0.7956E 02	0.7953E 02	0.7952E 02
0.4500E 00	0.8550E 00	0.8442E 02	0.8483E 02	0.8470E 02	0.8474E 02
0.5000E 00	0.9500E 00	0.8631E 02	0.8574E 02	0.8598E 02	0.8582E 02